CA Endevor® Software Change Manager

Installation Guide

Version 17.0.00

Second Edition

CA Technologies Product References

This document references the following CA Technologies products:

- CA ACF2™ for z/OS (CA ACF2)
- CA Endevor® Software Change Manager (CA Endevor SCM)
- CA Endevor® Software Change Manager CA Librarian® Interface (CA Endevor CA Librarian Interface)
- CA Endevor® Software Change Manager CA Panvalet® Interface (CA Endevor CA Panvalet Interface)
- CA Endevor® Software Change Manager Quick Edit (CA Endevor Quick Edit)
- CA Endevor® Software Change Manager Integration for the Natural Environment (CA Endevor Integration for the Natural Environment)
- CA Change Manager Enterprise Workbench (CA CMEW)
- CA Chorus™ Software Manager (CA CSM)
- CA Top Secret ® for z/OS (CA Top Secret)

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Documentation Changes

Note: In PDF format, page references identify the first page of the topic in which a change was made. The actual change may appear on a later page.

Version 17, Second Edition

- How the Installation Process Works (see page 19): Updated to refer to different versions of CA CSM.
- Installing Your Product Using CA CSM (see page 33): Updated to refer to the CA CSM documentation.
- Changes to the Chapter "Integration for the Natural Environment":
  - Install User Exits (see page 209)— Updated to add option L for the UEXIT00 variable +EINE-SYSTRAN-SYSOBJH.
  - Install the Processors to Use the Natural SYSOBJH Utility (see page 231)— Updated to add a step to modify the Generate processor PNATGEN.

Version 16.0, Second Edition

For Version 16.0 PTF RO65655, December 2013:

- Install User Exits (see page 209)— Updated to add the CA Endevor Integration for the Natural Environment UEXIT00 exits +EINE-RESTRICT-JOBCARDS, +EINE-LOGON-EXIT-ENABLED, +EINE-DSPLY-MENU-PACKAGES, +EINE-DSPLY-MENU-SAVEDSCL, +EINE-DSPLY-MENU-DDM, and +EINE-DSPLY-MENU-PREDICT. Also, added the logon exit, UEXIT01. Updated +EINE-SYSTRANS-SYSOBJH to add the optional value of F for SYSOBJH Transfer format. Also, added the new Package Execution Window parameters: +EINE-DFLT-EW-FROM-DATE, + EINE-DFLT-EW-FROM-TIME, +EINE-DFLT-EW-TO-DATE, and +EINE-DFLT-EW-TO-TIME.

Version 16.0

- Installing Your Product from Pax-Enhanced ESD— Replaced with the chapter Installing Your Product Using Pax ESD or DVD (see page 35).
- Installing Your Product from Tape— Obsolete topic removed from the guide.
- Software Requirements for Eclipse-Based UI (see page 195)— Updated to refer to the CA Endevor Quick Edit option as a requirement for the Eclipse-Based UI.
- Appendix "Web Services"— Updated, restyled, and moved to the scenario-based knowledge document *How to Enable Web Services,* which can be found in the *Scenario Guide*.
- Install the Adabas Files (see page 204)— Updated regarding running concurrent releases.

- Edit, Assemble, and Relink the NATPARM Module (see page 205)— Updated to add a step if you are using local buffer pools.
- Install User Exits (see page 209)— Updated to add the CA Endevor Integration for the Natural Environment UEXIT00 exits +EINE-MAX-ADD-RETRIEVE-PER-JOB, +EINE-STARTUP-CMD. Also added the +EINE-DFLT-xxxxxx exits that are used to establish site specific initial default values presented on the EINE Foreground and Batch Processing screens for the CA Endevor SCM Action Options.
- How to Configure a Natural Lifecycle in CA Endevor SCM (see page 220)— Updated to add a step to activate CA Endevor Integration for the Natural Environment in ENCOPTBL.
- Edit, Reassemble, and Relink ENCOPTBL (see page 222)— Added this topic.
- How to Add Site Symbolics to Associate Natural with CA Endevor SCM (see page 222)— Updated to specify that there is a maximum limit of 99 sets of stage and system site symbolics.
- Appendices "Implementation Considerations," "Implementing CA Endevor SCM," and "Frequently Asked Questions"— Obsolete appendices removed.

Release 15.1

- CA Common Services Requirements (see page 23)— Updated to add a note that if your site is running CAIENF r12 or higher, the database is not needed for Concurrent Action Processing or Web Services.
- Install User Exits— Added the following to the list of parameters for the Initialization Exit, UEXIT00, for the CA Endevor Integration for the Natural Environment: +EINE-MAX-ADD-RETRIEVE-PER-JOB.
- How to Enable the CACCI Spawn Facility—Updated to change a reference to member WSEWSCCI in a note about CCI definitions for Web Services.
- Log delta format—Updated to add HFS and binary files as appropriate for this format.
- Web Services—This chapter is updated in the following topics to provide information about configuring the Apache Tomcat server provided with CA Common Services.
  - How to Install and Configure Web Services
  - Determine Configuration Parameters
  - Establish Environmental Variables
  - Run the Installation and Configuration Script
- How to Enable CA Common Services for Web Services— Updated to add that if your site is running CAIENF r12 or higher, the database is not needed for Web Services.
- Deploy Web Services to an Existing Tomcat Server— Added to describe this procedure.

- Install the Eclipse-Based UI on a Client Computer (see page 196)— Updated to include Eclipse 3.7.

Version 15.0 Second Edition

- Installation Tasks for the Systems Programmer (see page 16)—Updated for the Software Configuration Service.
- How the Installation Process Works—Updated for the Software Configuration Service.
- How to Configure a Product—Added for the Software Configuration Service.
- Starting Your Product (see page 57)—Updated for the Software Configuration Service. This chapter was previously titled "Configuring Your Product."

Version 15.0

- Installation Tasks for the Systems Programmer (see page 16)—Updated for Web Services.
- How to Install and Configure Product Options (see page 21)—Added to refer to the appendices for product options.
- Installation Checklist (see page 92)—Updated to replace references to AUTHLIB.
- How to Edit the CSIQJCL Data Set—Updated the note about the file name changes introduced in Version 14.0 for CA CSM.
- Sample Processors—Updated to remove obsolete references to sample processors.
- @@UPDJCL Parameters Worksheet (see page 136)—Updated to indicate that the IPRFX, IQUAL, and UPRFX values must match the values specified for the base CA Endevor SCM installation or runtime libraries. Also, updated the recommended data set file type for DSN@TYPE to PDSE. Deleted the obsolete parameter UQUAL.
- 2-Verify the CA Endevor SCM Table Data Set (see page 137)—Updated to correct a reference to the library that stores tables from CSIQAUTH to CSIQAUTU.
- 3-Process the CA Endevor SCM Tables (see page 143)—Updated to correct a reference to the library that stores tables from CSIQAUTH to CSIQAUTU.
- 8-Define the DEV, QA, and PRD APPL@SYS Files (see page 145)—Updated to clarify which delta libraries are defined by @JOB0070.
- 9-Define the DEV, QA, and PRD COMM@SYS Files (see page 145)—Updated to clarify which delta libraries are defined by @JOB0080.
- 15-Copy the Table Load Modules to the CSIQAUTU Library (see page 151)—Updated to correct a reference to iprfx.iqual.CSIQAUTU.
- 2-Create an Additional System (see page 161)—Updated to correct references to SCLDE23, SCLDE24, and SCLDE25.
- CA Endevor Quick Edit (see page 179)—Added this appendix for CA Endevor Quick Edit.
- Web Services (see page 193)—Added this appendix for Web Services.

- Eclipse-Based UI (see page 195)—Added this appendix for Eclipse-Based UI.
- CA Endevor Integration for the Natural Environment (see page 199)—Added this appendix for CA Endevor Integration for the Natural Environment. Previously, the content of this appendix was included in the *CA Endevor Integration for the Natural Environment Options Guide*, which has been discontinued. The following substantive documentation changes were made to this text:
  - Install User Exits–Added more information about the variables for the CA Endevor Integration for the Natural Environment installation exit, EINE_UEXIT00.
  - Install the Adabas (see page 204)–Added an important note for clients who had previously installed a earlier release.
  - Create a Default CA Endevor SCM Procedure for Package Execution (see page 218)–Added this procedure, which describes how to create default JCL for package execution using the text module NDVPROCP in the EINELIB library.
  - Install the Processors (see page 230)–Updated to indicate that the SYSOBJH utility is the default for storing objects in CA Endevor SCM.
    - Install the Processors to Use the Natural SYSOBJH Utility (see page 231)–Updated to indicate that SYSOBJH is the default.
    - Install the Processors to Use the Natural SYSTRANS Utility (see page 231)–Updated to indicate that you can configure SYSTRANS, instead of using SYSOBJH to store objects in CA Endevor SCM.
- Japanese Panels and Messages (see page 245)—Added this appendix for the Japanese panels and messages option.
- Installation Data Set Names (see page 249)Edit, Reassemble, and Relink ENCOPTBL— (see page 222)Added this appendix to list the data set names used for releases prior to [assign the value for ENDV in your book] r14 and the names used beginning with r14.

Contents

Chapter 1: Overview

This section contains the following topics:

Audience

We strongly recommend that you read this entire document before starting an installation.

To install CA Endevor SCM, you must have the following knowledge:

- You need a working knowledge of the mainframe environment, the z/OS mainframe operating system, the Time Sharing Option facility (TSO), the Interactive System Productivity Facility (ISPF), and the System Modification Program / Extended (SMP/E).
- You need a working knowledge of OS JCL, the standard IBM utilities, and how to use the ISPF/PDF editor.
- You must know how to administer VSAM catalog structures.
- You need a working knowledge of, and can execute, the IBM SMP/E utility program.
- You must have the proper security authority to define data sets and perform system authorization functions.

You may need to work with the following personnel:

- Systems programmer, for z/OS and VTAM definitions.
- Storage administrator, for DASD allocations.
- Security administrator, for security configuration.
- CA Endevor SCM administrator, for the JCL variables to be defined for your site. For more information, see Installation Plan (see page 30).

Note: At many sites, the installation is performed by the systems programmer and the implementation is performed by the CA Endevor SCM administrator. For more information, see Implementation Tasks for the Administrator (see page 17).

Installation Tasks for the Systems Programmer

Note: The basic installation steps are described in How the Installation Process Works.

Important: Beginning with CA Endevor SCM r14, the product data set names were changed to make CA Endevor SCM compatible with CA CSM. For tables showing the new file names introduced with r14 and those files delivered before r14, see the appendix "Installation Data Set Names (see page 249)". This information is provided in case you are upgrading from CA Endevor SCM r12 or earlier.

The systems programmer usually completes the following steps in the installation process:

1. Confirm that your site meets the installation requirements.
2. Install and customize the CA Common Services components (see page 23) CA LMP and CAIRIM for product licensing and authorization.
3. (Optional) If you plan to use the Concurrent Action Processing (see page 29) (CAP) feature, install and customize CAIENF and the CA Common Communications Interface (CAICCI) for CAP.
4. (Optional) If you plan to use Web Services (see page 193), install and customize CAIENF and the CA Common Communications Interface (CAICCI) for Web Services.
5. Authorize NDVRC1 (see page 28).
6. Use one of the following methods to install CA Endevor SCM:
   - Install using CA CSM
   - Install from Pax-Enhanced ESD

   Note: If you are installing Web Services from ESD, the installation creates a SAMPJCL data set with a SCMAREAD file that mentions these members: SI31ALLU, SI32MKD, and SI33CSIU. If you are using USS data sets, these jobs are required for FMID CSIQF03 (Web Services). Execute these jobs after the SCM2CSI job, but before executing one of the receive jobs.

   - SI31ALLU – Allocates product USS Datasets.
   - SI32MKD – Allocates product USS Directories.
   - SI33CSIU – Customizes SMPE CSI for USS.

   Note: Regardless of the installation method you are using, the following steps must be completed.

7. Deploy and configure the delivered files and configure your installation environment as described in the chapter "Start Your Product (see page 57)". This includes the following tasks:
   a. Submit job BC1JJB03 to edit the CSIQJCL data set for your site (see page 63).
   b. Authorize the CSIQAUTU or CSIQAUTH libraries or copy the content to an existing authorized library (see page 69).

      Note: Existing installations that need to run their current release and the new release on the same system/LPAR need to verify that the libraries for the new release cannot be used by accident. For more information, see Concurrent Releases.
   c. Create a CA Endevor SCM environment for the new release under TSO (see page 75).
   d. Define CA Endevor SCM to GRS (if not already completed) (see page 72).
   e. Set up data set (see page 80) and functional security (see page 81).
8. Modify the TSO Command Limiting Facility to allow ISPTLIB commands (see page 31) (ACMQ, BC1PACMI, ESORT, EONLY, EM, QM, and so on). CA ACF2 for z/OS users only.

More Information

How to Install and Configure Product Options (see page 21)

Implementation Tasks for the Administrator

At many sites, the installation is performed by the systems programmer and the implementation is performed by the CA Endevor SCM administrator.

Note: The following implementation tasks are discussed in this guide. After completing these tasks, the product can be customized to meet the specific needs of your organization. For information about customization, see the *Administration Guide*.

1. The administrator needs to work with the Systems Programmer to complete the Installation Plan (see page 30) worksheets.

2. After installation, the following implementation tasks are required.

   - If you are installing CA Endevor SCM for the first time, we recommend that you use the Best Practice Implementation. You can use the Best Practice Implementation steps with minimum customization to set up a fully scalable software development environment that is complete and includes a lifecycle and inventory structure. For more information about the Best Practice Implementation, see the *Scenario Guide*.

     If you are not using the Best Practice Implementation, you can use the sample application (see page 83) to test your implementation of CA Endevor SCM and to gain a better understanding of the features CA Endevor SCM provides.

   - If you are an existing client and CA Endevor SCM is already installed and configured at your site, you can use the sample application (see page 83) or your existing data (see page 107) to test your installation of CA Endevor SCM.

More Information

How to Install and Configure Product Options (see page 21)

Post-Implementation Tasks for the Administrator

After installation, the CA Endevor SCM administrator usually performs the following steps:

1. Implement a Test Environment.

   To complete this step, do the following:

   a. Read the frequently asked questions.

   b. Read the information about how to test new releases with your data (see page 107).

   c. (Optional) Implement the sample application (see page 85).

2. Migrate the Existing Environment.

   Complete these steps to migrate your existing environment:

   a. Review the customization tables.

      Note: For more information, see Customization Tables in the *Administration Guide*.

   b. Recompile your exit programs.

c. Perform foreground validation of user table modifications to C1DEFLTS, ENDICNFG, ENCOPTBL, and SYMBOL TABLE as follows:

   1. Allocate file EN$TROPT under TSO using this command: TSO ALLOC FI (EN$TROPT) SYSOUT REUSE.

   2. Start CA Endevor SCM in foreground.

   3. Split the screen and review the file under your TSO session in the SDSF DA queue.

   4. Confirm that all selected options and table settings are correct.

d. Perform batch validation of user table modifications (C1DEFLTS, ENDICNFG, ENCOPTBL, and SYMBOL TABLE).

   - Submit a batch SCL job with the following additional DD statement: //EN$TROPT DD SYSOUT=*
   - After the job completes, view the output in SDSF and perform the same foreground validation as listed in the previous step.

e. If applicable, activate security in the C1DEFLTS table and reassemble your existing security tables (BC1TNEQU or Native Security tables) using the macro library (*ipfrx.iqual*.CSIQOPTN). For more information, see the *Security Guide*.

How the Installation Process Works

CA Technologies has standardized product installations across all mainframe products. Installation uses the following process:

- Acquisition—Transports the software to your z/OS system.
- Installation using SMP/E—Creates an SMP/E environment and runs the RECEIVE, APPLY, and ACCEPT steps. The software is untailored.
- (For CA CSM Release 5.1 and earlier only) Deployment—Copies the target libraries to another system or LPAR.

  Note: This step is optional for CA CSM Version 6.0. For more information, see the scenario *Configuring Products Using CA CSM* that is available in the CA CSM Version 6.0 bookshelf at http://ca.com/support.
- Configuration—Creates customized load modules, bringing the software to an executable state.
- (For staging system configurations in CA CSM Version 6.0 only) Deployment—Makes configured run-time libraries available to a remote location where that software can be activated, bringing it to an executable state.

CA Chorus™ Software Manager (CA CSM) - formerly known as CA Mainframe Software Manager™ (CA MSM) - is an intuitive web-based tool that can automate and simplify many CA Technologies product installation activities on z/OS systems. This application also makes obtaining and applying corrective and recommended maintenance easier. A web-based interface enables you to install and maintain your products faster and with less chance of error. As a best practice, we recommend that you install mainframe products and maintenance using CA CSM. Using CA CSM, someone with limited knowledge of JCL and SMP/E can install a product.

Note: If you do not have CA CSM, you can download it from the Download Center at http://ca.com/support. Follow the installation instructions in the CA Chorus Software Manager documentation bookshelf on the CA Chorus Software Manager product page.

You can also complete the standardized installation process manually using pax files that are downloaded from http://ca.com/support or a product DVD.

To install your product, do the following tasks:

1. Prepare for the installation by confirming that your site meets all installation requirements.
2. Verify that you acquired the product using one of the following methods:
   - Download the software from http://ca.com/support using CA CSM.
   - Download the software from http://ca.com/support using Pax-Enhanced Electronic Software Delivery (Pax ESD).
   - Order a product DVD. To do so, contact your account manager or a CA Technologies Support representative.
3. Perform an SMP/E installation using one of the following methods:
   - If you used CA CSM to acquire the product, start the installation process from the SMP/E Environments tab in CA CSM.
   - If you used Pax ESD to acquire the product, you can install the product in the following ways:
     - Install the product manually.
     - Complete the SMP/E installation using the Add Product option in CA CSM.
   - If you used a DVD, install the product manually.

   Note: If a CA Recommended Service (CA RS) package is published for your product, install it before proceeding.
4. (For CA CSM Release 5.1 and earlier only) Deploy the target libraries.

   Note: This step is optional for CA CSM Version 6.0. For more information, see the scenario *Configuring Products Using CA CSM* that is available in the CA CSM Version 6.0 bookshelf at http://ca.com/support.

5. Configure your product using CA CSM or manually.
6. (For staging system configurations in CA CSM Version 6.0 only) Deploy configured run-time libraries, and activate your product.

   Note: Configuration is considered part of starting your product (see page 57).

How to Install and Configure Product Options

After installing and configuring the product, additional tasks are required to set up product options and optional components. Some options have additional software and hardware requirements. Details about each of the following options are discussed in appendices of this guide.

- **CA Endevor Quick Edit** – For more information, see the appendix CA Endevor Quick Edit (see page 179).
- **Web Services** – For more information, see the standalone scenario-based knowledge document *How to Enable Web Services.* This scenario is also included in the new *Scenario Guide,* which is available in HTML or PDF format.
- **Eclipse-Based UI** – For more information, see the appendix Eclipse-Based UI (see page 195).
- **CA Endevor Integration for the Natural Environment** – For more information, see the appendix CA Endevor Integration for the Natural Environment. (see page 199)
- **Japanese Panels and Messages** – For more information, see the appendix Japanese Panels and Messages (see page 245).

Chapter 2: Preparing for Installation

This section describes what you need to know and do before you install the product.

This section contains the following topics:

Software Requirements

The following software is required for CA Endevor SCM:

- IBM Supported release of z/OS 1.9 or above and corresponding version of TSO ISPF
- IBM SMP/E utility

CA Common Services Requirements

CA Endevor SCM uses CA Common Services for product licensing and authorization. CA Common Services is a group of system services that protect your investment in software products by helping to manage your data center more efficiently.

The CA Common Services components used by CA Endevor SCM include CAIRIM, CA LMP, and CAIENF. These services help you run CA Endevor SCM.

CAIENF and CAICCI are required only if you want to use the Concurrent Action Processing feature, the Web Services component, or the CA CM Enterprise Workbench product. For more information about CA Common Services for Web Services, see How to Enable CA Common Services for Web Services.

Note: If other CA products are installed at your site, some of these services may already be installed.

Note: If your site is running CAIENF r12 or higher, a database is not required for Concurrent Action Processing or Web Services.

CAIRIM

CA Endevor SCM uses the CA Common Services Resource Initialization Manager (CAIRIM) component for product license authorization. CAIRIM routines are grouped under CA z/OS Dynamic Service Code S910.

Important! CAIRIM is mandatory for CA Endevor SCM. It must be installed and started with CA Endevor SCM within 30 minutes of IPL time. CAIRIM is part of CA Common Services for z/OS. For more information about the features and associated utilities of CAIRIM, see the *CA Common Services* for *z/OS Getting Started* guide.

CA LMP

CA Endevor SCM requires the proper level of the CA License Maintenance Program (LMP) to initialize correctly. CA LMP, which provides a standardized, automated approach to the tracking of licensed software, is provided as an integral part of CAIRIM.

The CA License Management Program uses common real-time enforcement software to validate the user's configuration. CA LMP reports on license, usage, and financial activities of CA Endevor SCM. The routines that accomplish this are integrated into the CA z/OS dynamic service code S910 (the CAIRIM service).

Some of the features of CA LMP include:

- Common key data set can be shared among many CPUs.
- *Check digits* are used to detect errors in transcribing key information.
- Execution keys can be entered without affecting any CA software solution already running.
- No special maintenance requirements exist.

Note: For more information about defining the CA LMP execution key to the CAIRIM parameters, see the *CA Common Services for z/OS Getting Started*.

The LMP Key Certificate

Examine the CA LMP Key Certificate you received with your CA Endevor SCM installation or maintenance cartridge. Your certificate contains the following information.

Product Name

The trademarked or registered name of CA Endevor SCM as licensed for the designated site and CPUs.

Product Code

A two-character code that corresponds to one or more CA Endevor SCM features.

Supplement

The reference number of your license for the particular CA Endevor SCM feature, if applicable.

CPU ID

The code that identifies the specific CPU for which the installation of CA Endevor SCM is valid.

Execution Key

An encrypted code required by CA LMP for product initialization. Also referred to as the LMP CODE.

Expiration Date

The date your license for CA Endevor SCM expires in the format *ddmmmyy*.

Contact Technical

The name of the designated technical contact at your site, who is responsible for installation and maintenance of CA Endevor SCM. This is the person to whom CA addresses all CA LMP correspondence.

MIS Director

The name of the Director of MIS, or the person who performs this function at your site. If the title, but not an individual's name, is indicated on the certificate, you should supply the actual name when correcting and verifying the certificate.

CPU Location

The address of the building in which the CPU is installed.

LMP Execution Keys

You must add the CA LMP Execution Keys provided on the Key Certificate to the CAIRIM parameters to ensure proper initialization of CA Endevor SCM. To define a CA LMP Execution Key, modify member KEYS in CALPPOPTION.

The following information shows the syntax for statements in the KEYS member and describes the individual parameters:

```
PROD(pp) DATE(ddmmmyy) CPU(tttt-mmmm/ssssss) LMPCODE(kkkkkkkkkkkkkkkk)
```

pp

The two-character product code. This required code matches the product code already in use by CAIRIM for any earlier versions of CA Endevor SCM, if applicable.

ddmmmyy

The licensing agreement expiration date.

tttt-mmmm

The CPU type and model on which the product is to run. If the CPU type or model requires less than four characters, use blanks for the unused characters. This is a required parameter.

ssssss

The serial number on which the product is to execute. This is a required parameter.

kkkkkkkkkkkkkkkk

The Execution Key required to run the product. This key is provided on the Key Certificate shipped with each CA software solution.

The following is an example LMP KEYS entry. The product code and execution key values will differ in your installation. Note the trailing blank in the CPU model number.

```
PROD(ZZ) DATE(08DEC99) CPU(3090-600 /370623) LMPCODE(0408A0423C0123EF)
```

Note: For more information about defining the CA LMP Execution Key to CAIRIM, see the *CA Common Services for z/OS Getting Started*.

Verification of the LMP Execution Keys Installation

Before enabling the test environment, ensure that all necessary CA Common Services LMP Execution Keys have been installed as described in the *CA Common Services for z/OS Getting Started*. You can use the same execution key being used for your production environment; a new key is not required when upgrading to a newer release of the product.

Some users may not be licensed for all features of the product. The demonstration job stream that is part of this phase uses facilities from all features of CA Endevor SCM.

Ensure that a valid CA LMP Execution Key, supplied by CA, has been specified to the CA Common Services CAIRIM licensing component.

Note: If you do not have a full feature execution key, you can obtain a temporary one by contacting the CA LMP support group at http://ca.com/support.

LMP Key Requirements

The CA License Management Program (CA LMP) tracks licensed software in a standardized and automated way. CA LMP uses common real-time enforcement software to validate the user configuration. CA LMP reports on activities that are related to the license, usage, and financials of CA Technologies products.

Your product is licensed with an LMP key. You acquire the LMP key with one of the following methods:

- From your product media
- With Pax ESD
- From http://ca.com/support

Note: For more information about LMP keys, see the CA Common Services for z/OS documentation.

Verify CA Common Services are Running

The CA Common Services components CAIRIM, CAIENF, and CAICCI must be installed and running on your mainframe. CAIENF and CAICCI are required only if you want to use the Concurrent Action Processing feature, the Web Services component, or the companion product CA CMEW. If they are not already installed and running, you must install these services according to the instructions in the *CA Common Services for z/OS Getting Started.*

To determine whether CAIENF and CAICCI are installed and running

1. Go to the SDSF status display and enter the prefix ENF*.

   Note: If you do not see ENF* active on your system, contact your systems programmer to start it.

2. If the job is running, select it and search for (Find) CAICCI within the job.
   - If you find CAICCI, then CAIENF and CAICCI are running.
   - If you do not find CAICCI, then check your site's CAI.CAIPROC library to see if the ENF member has been modified to run CAICCI at your site.

     If the ENF member does not exist, then contact your CA Common Services administrator (or systems programmer) to determine the status of CAIRIM, CAIENF, and CAICCI on your system.

Security Requirements

The following security requirements are needed to enable you to complete the tasks in this guide:

- You must have the proper security authority to define data sets and perform system authorization functions.
- The TSO profile prefix must be set to the user ID or to NOPREFIX.

Storage Requirements

Assuming your site uses 3390-type disk devices, DASD space requirements for the installation-related data sets for CA Endevor SCM are approximately 50 cylinders.

Important! All SPACE parameters are based on 3390 devices. If your site uses another type of device, you must adjust them accordingly.

Note: DASD space requirements for the implementation data sets will vary from site to site, depending upon the number of elements placed under control and various implementation options, such as the package feature, processor design, number of lifecycle stages, and so on.

NDVRC1 Authorization

Identify the CA Endevor SCM load module name NDVRC1 to the authorized TSO program, command, and service facility list. This process varies depending on the TSO environment in which you are operating. If you have performed this procedure for a previous release of CA Endevor SCM, you do not need to repeat it.

To authorize NDVRC1, add the load module name NDVRC1 to the AUTHPGM, AUTHCMD, and AUTHTSF sections of member IKJTSO00, in SYS1.PARMLIB.

Note: For more information about NDVRC1 authorization, see the IBM TSO/E documentation.

Activate the Changes

When you have added NDVRC1 to the appropriate lists and made the necessary changes to LNKLST*nn* and IEAAPF*nn*, you can use the PARMLIB and SETPROG commands to dynamically activate the changes or re-IPL the system.

How to Enable Concurrent Action Processing

The ENF procedure for CAICCI services enables the Concurrent Action Processing feature. Because Concurrent Action Processing uses the CAICCI SPAWN facility, you must append a Concurrent Action Processing spawn parameter file definition to the SPNPARMS DD in the ENF procedure. You can find the ENF procedure in the initial CAI.CAIPROC library. However, your system administrator may have copied the procedure to the system PROCLIB.

To enable Concurrent Action Processing, do the following:

1. Append a Concurrent Action Processing spawn parameter file definition named SPNPARMS DD to the ENF procedure.

   ```
   //SPNPARMS DD DISP=SHR,DSN=&OPTLIB(&SPNPAR1)
   //         DD DISP=SHR,DSN=&OPTLIB(&SPNPARn)
   ```

   SPNPARn

   A symbolic for a CAIENF parameter member, where *n* is replaced by the next available sequential number in the list of parameter files.

2. Customize the SPNPARn parameters by editing the CAICCI SERVICE and PROCESS statements in the SPNPARn file.

   SERVICE statement

   Identifies the host application

   PROCESS statement

   Identifies the JCL procedure that executes Endevor

   ENDEVOR

   Specifies the name of the procedure associated with the CA Endevor SCM host application to be started by the CAICCI SPAWN service during Concurrent Action Processing. The ENDEVOR value is the name of the started task procedure that is initiated by a SPAWN request.

   Note: It is important to keep the statements in the exact format provided, maintaining spaces and column alignments. For complete information on defining and using CAICCI statements, see the CA Common Services documentation.

   Note: For sample CCI definitions for Concurrent Action Processing, see the member CAPCCI supplied in the installation source library *iprfx.iqual*.CSIQOPTN.

3. Customize the ENDEVOR Procedure. A sample ENDEVOR procedure is located in the CSIQJCL library delivered with CA Endevor SCM.

   a. After modification, copy it into a PROCLIB defined to JES. ENDEVOR is initiated as a started task.

   b. If you change the ENDEVOR member name, be sure to update the PROCNAME value in the CCI PROCESS statement of the SPNPARn file to reflect the new procedure name.

4. Make sure that a default Region ACID is defined, if required for your site's security system. Because ENDEVOR is initiated as a started task (STC), it is likely that a default Region ACID must be defined for ENDEVOR to your security system. Consult your security administrator for more information.
5. If this is your initial install of these CAICCI components, you do not need to IPL your system after installation. If this is a subsequent install, see the CA Common Services for z/OS documentation for more information.
6. (Optional) If you want to use Workload Management to control the resources used by the spawned STCs, we recommend the following definitions.
   a. For the Action Request Regions—The action request regions are the started tasks that are created to service requests concurrently. The started task procedure name is defined in the PROCNAME= parameter of the CCI PROCESS statement.

      Define the started tasks in a Service class that is below 1st Period TSO, and above the 1st Period of Regular Development Batch. This is an Importance of 2 or 3, depending on the definition of the 2 Service classes.
   b. For the Routing Region—The routing region is the batch job or CA CM Enterprise Workbench started task that has requested that actions be processed concurrently. These jobs can be in Service class SYSSTC, or a high-priority Service class such as STCHI. (Typically, STCHI is defined as Importance 2 with a Velocity of 60.)

Installation Plan

To ensure that you successfully install and configure CA Endevor SCM, organize and plan your installation carefully. Use the following tables and worksheets to review and record the information and values you need to use in the installation, and to record the estimated disk space requirements for your site.

- Variable Definitions Table (see page 113)

  Review the detailed definitions for several variables in the different installation JCL jobs.
- Installation Worksheet (see page 117)

  Record the information and values you need to install at your site.
- Master Control File Allocations for the Sample Application (see page 117)

  Provides allocation information for the CA Endevor SCM sample application.
- Disk Space Requirements Worksheet (see page 119)

  Estimate the disk space requirements for your site.

Concurrent Releases

You can install this release of your product and continue to use an older release in another SMP/E environment. If you plan to continue to run a previous release, consider the following points:

- When you install the product into an existing SMP/E environment, this installation deletes previous releases in that environment.
- If you acquired your product with Pax ESD, select different target and distribution zones for your new release from where your current release is installed. The new zones use different libraries than your current release.

  Note: CA CSM installs a product into a new SMP/E environment by default. You can select an existing SMP/E environment from your working set. For more information, see the online help that is included in CA CSM.
- Define DDDEF entries in your new zones to point SMP/E to the proper libraries for installation. Ensure that they point to the new release libraries.

Concurrent Release Considerations

In many cases, it is necessary to run two different releases of CA Endevor SCM at the same time. You can do this by adding your CSIQAUTH/CSIQAUTU and CSIQLOAD data sets to the respective STEPLIB and CONLIB statements in a test TSO logon procedure. Member SCMM@LIB in both the CSIQSENU and CSIQJCL installation libraries were tailored by BC1JJB03 of the installation process to point to your STEPLIB/CONLIB concatenations. The Skeleton JCL members installed imbed CSIQJCL member SCMM@LIB to point to your STEPLIB/CONLIB concatenations. Stand-alone JCL members include CSIQJCL member SCMM@LIB for your STEPLIB/CONLIB concatenations. The CSIQJCL member SCMM@LIB can be copied to a JES PROCLIB or a JCLLIB statement can be added to each stand-alone JCL member to point to the library where this member resides.

You *cannot* access both releases in the same TSO/ISPF session. After working with one release, before you can work with the other, you must log off TSO and log on again, making sure to change the logon procedures. If you do not change your logon procedures, a message indicates an incompatible C1DEFLTS Table.

ISP Command Table Considerations

CA Endevor SCM uses an ISP Command table (allocated under the ISPTLIB DD) for some commands (such as ESORT, EONLY, EPRINT, ACMQ, BC1PACMI, EMSGS, and so on). However, unless this facility is specifically enabled, it may be prevented from working by your site's security package. For example, CA ACF2 has a facility named TSO Command Limiting, which may need modification.

USS Space Requirements

Ensure that you have sufficient free space in the USS file system that you are using for Pax ESD to hold the directory that the pax command and its contents create. You need approximately 3.5 times the pax file size in free space.

If you do not have sufficient free space, you receive error message EDC5133I.

Chapter 3: Installing Your Products Using CA CSM

As a system programmer, your responsibilities include acquiring, installing, maintaining, deploying, and configuring CA Technologies mainframe products on your system.

CA CSM is an application that simplifies and unifies the management of your CA Technologies mainframe products on z/OS systems. As products adopt the CA CSM services, you can install your products in a common way according to industry best practices.

If you do not have CA CSM installed, download it from the Download Center at http://ca.com/support. This web page also contains links to the complete documentation for CA CSM.

You can use the following scenarios to guide you through the product installation process (see page 19) using CA CSM:

- Acquiring Products Using CA CSM
- Installing Products Using CA CSM
- Maintaining Products Using CA CSM
- Configuring Product Using CA CSM

These scenarios are available in the CA CSM Version 6.0 bookshelf at http://ca.com/support. For additional information about how to use CA CSM, use the online help.

Note: The CA CSM configuration process implements a Rapid Implementation configuration for CA Endevor SCM. Instead of using CA CSM to configure CA Endevor SCM, we recommend that you use the Best Practice Implementation. For more information about the Best Practice Implementation, see the *Scenario Guide*.

Chapter 4: Installing Your Product Using Pax ESD or DVD

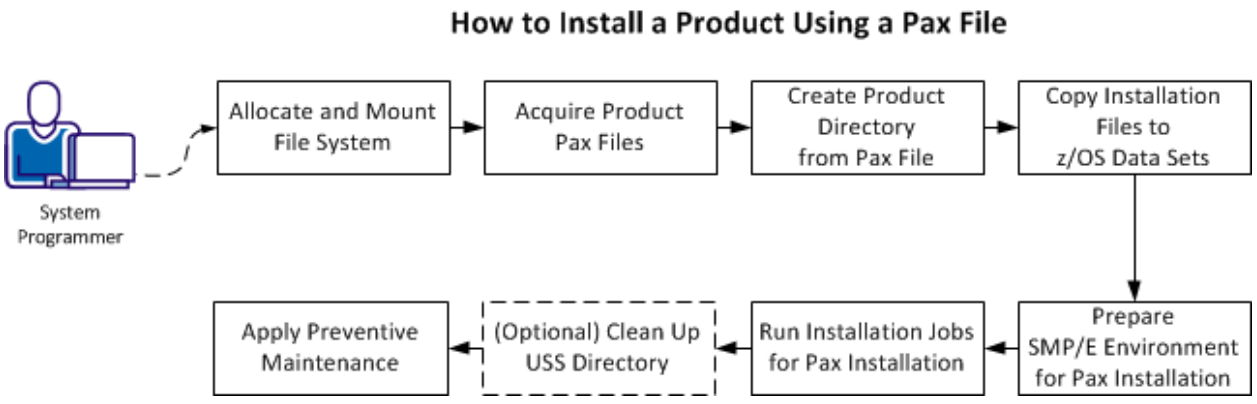
This section contains the following topics:

How to Install Your Product Using a Pax File

As a system programmer, your responsibilities include installing products on your mainframe system. With this option, you acquire a product pax file from http://ca.com/support or from a product DVD.

The DVD contains a folder that includes the pax file for the product. Product updates may have occurred after you acquired the product DVD. The files on the online site always have the most current product updates. To determine if you have the latest updates, go to http://ca.com/support and click Download Center.

You perform the following tasks to install a product with a pax file:



1. Allocate and mount the file system (see page 37).
2. Acquire the product pax files (see page 39).

3. Create a product directory from the pax file (see page 44).
4. Copy the installation files to z/OS data sets (see page 45).
5. Prepare the SMP/E environment for a pax installation (see page 47).
6. Run the installation jobs for a pax installation (see page 49).
7. (Optional) Clean up the USS directory (see page 50).
8. Apply preventive maintenance (see page 51).

USS Environment Setup

You need a UNIX System Services (USS) directory and a file system with adequate space to perform the following tasks:

- Receive product pax files from http://ca.com/support.
- Perform utility functions to unpack the pax file into MVS data sets that you can use to complete the product installation.

We recommend that you allocate and mount a file system that is dedicated to Pax ESD. The amount of space that you need for the file system depends on the following variables:

- The size of the pax files that you intend to download.
- Whether you plan to keep the pax files after unpacking them. We do not recommend this practice.

We recommend that you use one directory for downloading and unpacking pax files. Reusing the same directory minimizes USS setup. You need to complete the USS setup only one time. You reuse the same directory for subsequent downloads. Alternatively, you can create a directory for each pax download.

Important! Downloading pax files for the SMP/E installation as part of the Pax ESD process requires write authority to the UNIX System Services (USS) directories that are used for the Pax ESD process. In the file system that contains the Pax ESD directories, you also need free space approximately 3.5 times the pax file size to download the pax file and unpack its contents. For example, to download and unpack a 14 MB pax file, you need approximately 49 MB of free space in the file system hosting your Pax ESD directory.

Allocate and Mount a File System

The product installation process requires a USS directory to receive the pax file and to perform the unpack steps. We recommend that you allocate and mount a file system that is dedicated to the product acquisition and create the directory in this file system.

You can use the zSeries File System (zFS) or hierarchical file system (HFS) for product downloads.

This procedure describes how to perform the following tasks:

- Allocate a zFS or an HFS.
- Create a mount point in an existing maintenance USS directory of your choice.
- Mount the file system on the newly created mount point.

  Note: You must have either SUPERUSER authority, or the required SAF profile setting to allow you to issue the USS mount command for the file system.
- Optionally, permit write access to anyone in the same group as the person who created the directory.

  Important! USS commands are case-sensitive.

Follow these steps:

1. Allocate the file system by customizing one of the following samples to your site requirements:

   - On a zFS, use the following sample:

```
//DEFINE   EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//AMSDUMP  DD   SYSOUT=*
//SYSIN    DD *
  DEFINE CLUSTER ( +
  NAME(your_zFS_data_set_name) +
  STORAGECLASS(class) +
  LINEAR +
  CYL(primary secondary) +
  SHAREOPTIONS(3,3) +
  )
/*
//FORMAT   EXEC PGM=IOEAGFMT,REGION=0M,
// PARM=('-aggregate your_zFS_data_set_name -compat')
//SYSPRINT DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//STDOUT   DD   SYSOUT=*
//STDERR   DD   SYSOUT=*
//CEEDUMP  DD   SYSOUT=*
//*
```

- On an HFS, use the following sample:

```
//ALCHFS EXEC PGM=IEFBR14
//CAPAX  DD   DSN=yourHFS_data_set_name,
//            DISP=(NEW,CATLG,DELETE),UNIT=3390,
//            DSNTYPE=HFS,SPACE=(CYL,(primary,secondary,1))
```

The file system is allocated.

Note: Ensure that the zFS or HFS data set name that you use conforms to your data set naming conventions for USS file systems. If the allocation of the file system data set fails, it is because of environmental settings not allowing for the allocation. On an HFS, try using the ISPF 3.2 Data Set Utility to allocate your HFS data set.

2. Create a mount point for the file system. This example shows how to create a /CA/CAPAX directory in an existing directory, /u/maint. From the TSO OMVS shell, enter the following commands:

```
cd /u/maint/
mkdir CA
cd CA
mkdir CAPAX
```

Note: This document refers to this structure as *yourUSSpaxdirectory*.

The mount point is created.

3. Mount the file system by customizing one of the following samples to your site requirements:

- On a zFS, use the following sample:

```
MOUNT FILESYSTEM('your_zFS_data_set_name')
      MOUNTPOINT('yourUSSpaxdirectory')
      TYPE(ZFS)  MODE(RDWR)
      PARM(AGGRGROW)
```

- On an HFS, use the following sample:

```
MOUNT FILESYSTEM('your_HFS_data_set_name')
      MOUNTPOINT('yourUSSpaxdirectory')
      TYPE(HFS)  MODE(RDWR)
```

The file system is mounted.

4. (Optional) Set security permissions for the directory. You can use the chmod command to let other users access the Pax ESD directory and its files. For example, to allow write access to the Pax ESD directory for other users in your USS group, from the TSO OMVS shell, enter the following command:

```
chmod -R 775 /yourUSSpaxdirectory/
```

Write access is granted.

Note: For more information about the chmod command, see the IBM *z/OS UNIX System Services User Guide* (SA22-7802).

Acquire the Product Pax Files

To begin the CA Technologies product installation procedure, copy the product pax file into the USS directory that you set up.

Important! Downloading pax files for the SMP/E installation as part of the Pax ESD process requires write authority to the UNIX System Services (USS) directories that are used for the Pax ESD process. Also, you must have available USS file space before you start the procedures in this guide.

Use one of the following methods:

- Download the product pax file from http://ca.com/support to your PC (see page 40), and then upload it to your USS file system.

  If you download a zip file, you must unzip it before uploading to your USS file system.
- Download the pax files from http://ca.com/support directly to your USS file system (see page 40).
- Download the pax file from the product DVD to your PC, and then upload the pax files to your USS file system. (see page 43)

This section includes the following information:

- A sample batch job to download a product pax file from the CA Support Online FTP server directly to a USS directory on your z/OS system
- Sample commands to upload a pax file from your PC to a USS directory on your z/OS system

Important! The FTP procedures vary due to local firewall and other security settings. Consult your local network administrators to determine the appropriate FTP procedure to use at your site.

Ensure that sufficient free space is available in the USS file system that you are using to hold the product pax file. If you do not have sufficient free space, error messages similar to the following appear:

```
EZA1490I Error writing to data set
EZA2606W File I/O error 133
```

When the download finishes, the pax file size in your USS directory matches the value in the Size column for the corresponding pax file on the CA Technologies Products Download window.

Download Files to a PC Using Pax ESD

You can download product installation files from http://ca.com/support to your PC.

Follow these steps:

1. Log in to http://ca.com/support, and click Download Center.

   The Download Center web page appears.

2. Under Download Center, select Products from the first drop-down list, and specify the product, release, and gen level (if applicable), and click Go.

   The CA Product Download window appears.

3. Download an entire CA Technologies product software package or individual pax files to your PC. If you download a zip file, you must unzip it before continuing.

   Note: For traditional installation downloads, see the *Traditional ESD User Guide*. For information about download methods, see the Download Methods and Locations article. Go to http://ca.com/support, log in, and click Download Center. Links to the guide and the article appear under the Download Help heading.

Download Using Batch JCL

You download a pax file from http://ca.com/support by running batch JCL on the mainframe. Use the sample JCL attached to the PDF file as CAtoMainframe.txt (see page 42) to perform the download.

Important! The PDF version of this guide includes sample JCL jobs that you can copy directly to the mainframe. To access these jobs, click the paper clip icon at the left of the PDF reader. A window displaying attachments opens. Double-click a file to view a sample JCL. We recommend that you use the latest version of Adobe Reader for viewing PDF files.

Note: We recommend that you follow the preferred download method as described on http://ca.com/support. This JCL procedure is our preferred download method for users who do not use CA CSM. We also include the procedure to download to the mainframe through a PC in the next section.

Follow these steps:

1. Replace *ACCOUNTNO* with a valid JOB statement.
2. Replace *yourTCPIP.PROFILE.dataset* with the name of the TCP/IP profile data set for your system. Consult your local network administrators, if necessary.

   The job points to your profile.

3. Replace *YourEmailAddress* with your email address.

   The job points to your email address.

4. Replace *yourUSSpaxdirectory* with the name of the USS directory that you use for Pax ESD downloads.

   The job points to your USS directory.

5. Locate the product component to download on the CA Support Product Download window.

   You have identified the product component to download.

6. Click Download for the applicable file.

   Note: For multiple downloads, add files to a cart.

   The Download Method window opens.

7. Click FTP Request.

   The Review Download Requests window displays any files that you have requested to download.

   Note: We send you an email when the file is ready to download or a link appears in this window when the file is available.

8. Select one of the following methods:

   Preferred FTP

   Uses CA Technologies worldwide content delivery network (CDN). If you cannot download using this method, review the security restrictions for servers that company employees can download from that are outside your corporate network.

   Host Name: ftp://ftpdownloads.ca.com

   Alternate FTP

   Uses the original download servers that are based on Long Island, New York.

   Host Name: ftp://scftpd.ca.com for product files and download cart files and ftp://ftp.ca.com for individual solution files.

   Both methods display the host, user name, password, and FTP location, which you then can copy into the sample JCL.

   Note: The following links provide details regarding FTP: the FTP Help document link in the Review Download Requests window and the Learn More link available in the Download Methods window.

9. Submit the job.

   Important! If your FTP commands are incorrect, it is possible for this job to fail and still return a zero condition code. Read the messages in the job DDNAME SYSPRINT to verify the FTP succeeded.

   After you run the JCL job, the pax file resides in the mainframe USS directory that you supplied.

Example: CAtoMainframe.txt, JCL

The following text appears in the attached CAtoMainframe.txt JCL file:

```
//GETPAX   JOB (ACCOUNTNO),'FTP GET PAX ESD PACKAGE',
//          MSGCLASS=X,CLASS=A,NOTIFY=&SYSUID
//*********************************************************************
//* This sample job can be used to download a pax file directly from  *
//* CA Support Online to a USS directory on your z/OS system.         *
//*                                                                   *
//* When editing the JCL ensure that you do not have sequence numbers *
//* turned on.                                                        *
//*                                                                   *
//* This job must be customized as follows:                           *
//* 1. Supply a valid JOB statement.                                  *
//* 2. The SYSTCPD and SYSFTPD JCL DD statements in this JCL may be   *
//*    optional at your site. Remove the statements that are not      *
//*    required. For the required statements, update the data set     *
//*    names with the correct site-specific data set names.           *
//* 3. Replace "Host" based on the type of download method.           *
//* 4. Replace "YourEmailAddress" with your email address.            *
//* 5. Replace "yourUSSpaxdirectory" with the name of the USS         *
//*    directory used on your system for Pax ESD downloads.           *
//* 6. Replace "FTP Location" with the complete path                  *
//*    and name of the pax file obtained from the FTP location        *
//*    of the product download page.                                  *
//*********************************************************************
//GETPAX   EXEC PGM=FTP,PARM='(EXIT TIMEOUT 120',REGION=0M
//SYSTCPD  DD   DSN=yourTCPIP.PROFILE.dataset,DISP=SHR
//SYSFTPD  DD   DSN=yourFTP.DATA.dataset,DISP=SHR
//SYSPRINT DD   SYSOUT=*
//OUTPUT   DD   SYSOUT=*
//INPUT    DD   *
Host
anonymous YourEmailAddress
lcd yourUSSpaxdirectory
binary
get FTP_location
quit
/*
```

Download Files to Mainframe through a PC

You download the product installation files to your PC and transfer them to your USS system.

Follow these steps:

1. Download the product file to your PC using one of the following methods:
   - Pax ESD (see page 40). If you downloaded a zip file, first unzip the file to use the product pax files.
   - DVD. Copy the entire product software package (or individual pax files) to your PC.

   The pax file resides on your PC.

   Note: Do *not* change the format of the pax.Z.

2. Open a Windows command prompt.

   The command prompt appears.

3. Customize and enter the following FTP commands:

```
FTP mainframe
userid
password
bin
lcd C:\PC\folder\for\thePAXfile
cd /yourUSSpaxdirectory/
put paxfile.pax.Z
quit
exit
```

mainframe

Specifies the z/OS system IP address or DNS name.

userid

Specifies your z/OS user ID.

password

Specifies your z/OS password.

C:*PC\folder\for\thePAXfile*

Specifies the location of the pax file on your PC.

Note: If you specify a location that has blanks or special characters in the path name, enclose that value in double quotation marks.

yourUSSpaxdirectory

Specifies the name of the USS directory that you use for Pax ESD downloads.

***paxfile*.pax.Z**

Specifies the name of the pax file to upload.

The pax file is transferred to the mainframe.

Create a Product Directory from the Pax File

The pax command performs the following actions:

- Extracts the files and directories that are packaged within the pax file.
- Creates a USS directory in the same directory structure where the pax file resides.
- Automatically generates a product and level-specific directory name.

Set the current working directory to the directory containing the pax file, and create a directory in your USS directory by entering the following command:

```
pax -rvf pax-filename
```

Use the sample JCL that is attached to the PDF file as Unpackage.txt (see page 45) to extract the product pax file into a product installation directory.

Important! The PDF version of this guide includes sample JCL jobs that you can copy directly to the mainframe. To access these jobs, click the paper clip icon at the left of the PDF reader. A window displaying attachments opens. Double-click a file to view a sample JCL. We recommend that you use the latest version of Adobe Reader for viewing PDF files.

Follow these steps:

1. Replace *ACCOUNTNO* with a valid JOB statement.
2. Replace *yourUSSpaxdirectory* with the name of the USS directory that you use for product downloads.

   The job points to your specific directory.
3. Replace *paxfile.pax.Z* with the name of the pax file.

   The job points to your specific pax file.
4. Submit the job.

   The job creates the product directory.

Note: If the PARM= statement exceeds 71 characters, uncomment and use the second form of UNPAXDIR instead. This sample job uses an X in column 72 to continue the PARM= parameters to a second line.

Example: JCL File, Unpackage.txt, to Customize

The following text appears in the attached Unpackage.txt JCL file:

```
//ESDUNPAX JOB (ACCOUNTNO),'UNPAX PAX ESD PACKAGE',
// MSGCLASS=X,CLASS=A,NOTIFY=&SYSUID
//*********************************************************************
//* This sample job can be used to invoke the pax command to create   *
//* the product-specific installation directory.                      *
//*                                                                   *
//* This job must be customized as follows:                           *
//* 1. Supply a valid JOB statement.                                  *
//* 2. Replace "yourUSSpaxdirectory" with the name of the USS         *
//*    directory used on your system for Pax ESD downloads.           *
//* 3. Replace "paxfile.pax.Z" with the name of the pax file.         *
//* NOTE: If you continue the PARM= statement on a second line,       *
//*       start entering characters in column 16 and make sure        *
//*       the 'X' continuation character is in column 72.             *
//*********************************************************************
//UNPAXDIR EXEC PGM=BPXBATCH,
// PARM='sh cd /yourUSSpaxdirectory/; pax -rvf paxfile.pax.Z'
//*UNPAXDIR EXEC PGM=BPXBATCH,
//* PARM='sh cd /yourUSSpaxdirectory/; pax                            X
//*             -rvf paxfile.pax.Z'
//STDOUT DD SYSOUT=*
//STDERR DD SYSOUT=*
```

Copy Installation Files to z/OS Data Sets

Use this procedure to invoke the SMP/E GIMUNZIP utility to create MVS data sets from the files in the product-specific directory.

The file UNZIPJCL in the product directory contains a sample job to GIMUNZIP the installation package. You edit and submit the UNZIPJCL job to create z/OS data sets.

Follow these steps:

1. Locate and read the product readme file or installation notes, if applicable, which resides in the product-specific directory that the pax command created. This file contains the product-specific details that you require to complete the installation procedure.

   You have identified the product-specific installation details.

2. Use ISPF EDIT or TSO ISHELL to edit the UNZIPJCL sample job. You can perform this step in one of the following ways:

   - Use ISPF EDIT. Specify the full path name of the UNZIPJCL file.
   - Use TSO ISHELL. Navigate to the UNZIPJCL file and use the E line command to edit the file.

   The job is edited.

3. Change the SMPDIR DD PATH to the product-specific directory created by the pax command.

   Your view is of the product-specific directory.

4. If ICSF is not active, perform the following steps:

   a. Change the SMPJHOME DD PATH to your Java runtime directory. This directory varies from system to system.

   b. Perform one of the following steps:

      - Change the SMPCPATH DD PATH to your SMP/E Java application classes directory, typically /usr/lpp/smp/classes/.
      - Change HASH=YES to HASH=NO on the GIMUNZIP parameter.

   One of the following occurs: ICSF is active or you are using Java.

5. Change all occurrences of y*ourHLQ* to the high-level qualifier (HLQ) for z/OS data sets that the installation process uses. We suggest that you use a unique HLQ for each expanded pax file to identify uniquely the package. Do *not* remove CAI after *yourHLQ*. Do *not* use the same value for y*ourHLQ* as you use for the SMP/E RELFILEs.

   All occurrences of y*ourHLQ* are set to your high-level qualifier for z/OS data sets.

6. Submit the UNZIPJCL job.

   The UNZIPJCL job completes with a zero return code. Messages GIM69158I and GIM48101I in the output and IKJ56228I in the JES log are acceptable.

   GIMUNZIP creates z/OS data sets with the high-level qualifier that you specified in the UNZIPJCL job. You use these data sets to perform the product installation. The pax file and product-specific directory are no longer needed.

   Note: For more information, see the IBM *SMP/E for z/OS Reference (SA22-7772)*.

Prepare the SMP/E Environment for a Pax Installation

The following steps describe the process to install products using native SMP/E JCL:

1. Download external HOLDDATA.
2. Allocate product data sets and SMP/E data sets.
3. Create an SMP/E environment.
4. Receive base functions and HOLDDATA.
5. Download and RECEIVE PTFs from http://ca.com/support.
6. Run an SMP/E APPLY CHECK operation.
7. Apply base functions using SELECT GROUPEXTEND.
8. Run an SMP/E ACCEPT CHECK operation.
9. Accept base functions using SELECT GROUPEXTEND.
10. Configure the product according to your site requirements.

Note: Steps 1 through 3 of this process are documented in detail in this section. Steps 4 through 9 are documented in the section describing how to run installation jobs for a Pax installation. If applicable to your product, Step 10 is documented in the section describing starting your product.

The members that are used in this procedure prepare the data sets, initialize the zones, and create the DDDEFs for your product.

Establishing a hierarchical file system (HFS) may be required as part of the product installation or required as a feature of the product.

For information about the members, see the comments in the JCL.

Follow these steps:

1. Customize the macro SCMSEDIT with your site-specific information and then copy the macro to your SYSPROC location. Replace the rightmost parameters for each ISREDIT CHANGE command. Each time you edit an installation member, type SCMSEDIT on the command line, and press Enter to replace the defaults with your specifications.

   The macro is ready to customize the *yourHLQ*.SAMPJCL members.

   Note: Set the DASD HLQ to the same value specified for *yourHLQ* within the JCL that is used to unzip the pax file.

   Note: The following steps include instructions to execute the SCMSEDIT macro each time you open a new SAMPJCL member. To edit all SAMPJCL members simultaneously, read and follow the instructions in the SCMAREAD member, and submit the SCMEDALL member.

2. Open the SAMPJCL member SCM1HOLD in an edit session and execute the SCMSEDIT macro from the command line.

   SCM1HOLD is customized.

3. Submit SCM1HOLD.

   This job downloads the error and FIXCAT HOLDDATA from http://ca.com/support.

4. Open the SAMPJCL member SCM2ALL in an edit session and execute the SCMSEDIT macro from the command line.

   SCM2ALL is customized.

5. Submit SCM2ALL.

   This job produces the following results:

   - The target and distribution data sets for your product are created.
   - Unique SMPLTS, SMPMTS, SMPSCDS, and SMPSTS data sets for this target zone are created.

6. If your product requires a USS file system or if you want to install a feature of the product that requires a USS file system, allocate and mount the file system:

   Note: You can customize the supplied HFS JCL to zFS, if your site requires it.

   a. Open the SAMPJCL member *ccc2*ALLU in an edit session and execute the SCMSEDIT macro from the command line.

      Note: All instances of *ccc* in this section indicate a three-character component code based on the FMID.

      *ccc2*ALLU is customized.

   b. Submit *ccc2*ALLU.

      This job allocates your HFS or zFS data sets.

   c. Open the SAMPJCL member *ccc3*MKD in an edit session and execute the SCMSEDIT macro from the command line.

      *ccc3*MKD is customized.

   d. Submit *ccc3*MKD.

      This job creates all directories and mounts the file system.

7. Open the SAMPJCL member SCM3CSI in an edit session and execute the SCMSEDIT macro from the command line.

   SCM3CSI is customized.

8. Submit SCM3CSI.

   This job produces the following results:

   - The CSI data set is defined.
   - The SMPPTS and SMPLOG data sets are allocated.

- The global, target, and distribution zones are initialized.
- The DDDEF entries for your product are created.
- The DDDEFs for the required SMP/E data sets are created.

9. If your product requires HFS or if you want to install a feature of the product that requires HFS, add the DDDEFS that are required for the file system to your SMP/E environment:

   a. Open the SAMPJCL member *ccc*3CSIU in an edit session and execute the SCMSEDIT macro from the command line.

      *ccc*3CSIU is customized.

   b. Submit *ccc*3CSIU.

      This job customizes the CSI by adding the DDDEFs associated with the directory.

Run the Installation Jobs for a Pax Installation

Submit and run these SAMPJCL members in sequence. Do not proceed with any job until the previous job has completed successfully.

Note: The following steps include instructions to execute the SCMSEDIT macro each time you open a new SAMPJCL member. To edit all SAMPJCL members simultaneously, read and follow the instructions in the SCMAREAD member, and submit the SCMEDALL member.

Follow these steps:

1. Open the SAMPJCL member SCM4RECD in an edit session, and execute the SCMSEDIT macro from the command line.

   SCM4RECD is customized.

2. Submit SCM4RECD to receive SMP/E base functions and error HOLDDATA.

   Your product is received and now resides in the global zone.

3. If an FMID was placed in error, download and receive PTFs (see page 51) from http://ca.com/support.

4. Open the SAMPJCL member SCM5APP in an edit session, and execute the SCMSEDIT macro from the command line.

   SCM5APP is customized.

5. Submit SCM5APP to apply SMP/E base functions with the CHECK option. If you find unresolved hold errors, we recommend that you note these errors and verify that resolving PTFs are applied before implementing products in production. Update the JCL to BYPASS the unresolved hold error IDs. After successful completion, rerun APPLY with the CHECK option removed.

   Your product is applied and now resides in the target libraries.

6. Open the SAMPJCL member SCM6ACC in an edit session, and execute the SCMSEDIT macro from the command line.

   SCM6ACC is customized.

7. Submit SCM6ACC to accept SMP/E base functions with the CHECK option. After successful completion, rerun APPLY with the CHECK option removed.

   Your product is accepted and now resides in the distribution libraries.

Clean Up the USS Directory

This procedure is optional. If you decide to perform the procedure, do so after you complete the installation process and when you do not need the installation files anymore.

To free file system disk space for subsequent downloads after downloading and processing the pax files for your CA Technologies product, we recommend removing the files from your USS directory and deleting unnecessary MVS data sets. You can delete the following items:

- Pax file
- Product-specific directory that the pax command created and all of the files in it
- SMP/E RELFILEs, SMPMCS, and HOLDDATA MVS data sets

  These data sets have the HLQ that you assigned in the UNZIPJCL job.

Note: Retain non-SMP/E installation data sets such as *yourHLQ*.INSTALL.NOTES for future reference.

Follow these steps:

1. Navigate to your Pax ESD USS directory.

   Your view is of the applicable USS directory.

2. Delete the pax file by entering the following command:

   `rm paxfile`

   paxfile

   Specifies the name of the CA Technologies pax file that you downloaded.

   The pax file is deleted.

3. Delete the product-specific directory by entering the following command:

   `rm -r product-specific_directory`

   product-specific_directory

   Specifies the product-specific directory that the pax command created.

   The product-specific directory is deleted.

Note: You can also use TSO ISHELL to navigate to the pax file and product-specific directory, and delete them using the D line command.

Apply Preventive Maintenance

Important! We strongly recommend that you use CA CSM to maintain your CA Technologies z/OS-based products. The procedure that is discussed in this section is fully automated when you use CA CSM.

CA Support Online at http://ca.com/support has maintenance and HOLDDATA published since the installation data was created. After the maintenance process completes, the product is ready to deploy.

Use this procedure during product installation and for ongoing preventive maintenance in non-installation use cases according to your maintenance strategy.

Note: To review the CA Technologies mainframe maintenance philosophy, see your *Best Practices Guide* or visit the CA Next-Generation Mainframe Management page.

This procedure directs you to use the CAUNZIP utility. The CAUNZIP utility processes ZIP packages directly on z/OS without the need for an intermediate platform, such as a Microsoft Windows workstation. If you are not familiar with this utility, see the *CA Common Services for z/OS Administration Guide*. This guide includes an overview and sample batch jobs. To use this utility, you must be running CA Common Services for z/OS Version 14.0 with PTF RO54887 or CA Common Services for z/OS Release 14.1 with PTF RO54635 and RO58216. These PTFs are included in CA Common Services for z/OS Release 14.1 at the S1401 Service Update level.

Follow these steps:

1. Check the Download Center at http://ca.com/support for PTFs that have been published since this release was created. If the base release was created recently, no PTFs will have been published yet. If PTFs exist, add published solutions for your product to your Download Cart, and click Checkout.
2. Specify that you want a complete package.

   When processing completes, a link appears on the Review Download Requests page. You also receive an email notification.

3. Click the Alternate FTP link for your order to obtain FTP login information and the ZIP file location. Download the ZIP file into a USS directory on your z/OS system.

4. Run the CAUNZIP utility.

   CAUNZIP unzips the package of published solutions and creates a SMPNTS file structure that the SMP/E RECEIVE FROMNTS command can process. For sample JCL to run the utility that is located in *yourHLQ*.CAW0JCL(CAUNZIP), see the *CA Common Services for z/OS CAUNZIP Administration Guide*. After execution completes, the ZIPRPT data set contains the summary report. The summary report does the following:

   - Summarizes the content of the product order ZIP file.
   - Details the content of each data set and the z/OS UNIX files produced.
   - Provides a sample job to receive the PTFs in your order.

5. Review the sample job that is provided in the CAUNZIP output ZIPRPT file. Cut and paste the JCL into a data set, specify your SMP/E CSI on the SMPCSI DD statement and submit the job to receive the PTFs in your order.

6. Verify that you have the values from the base installation in the SCMSEDIT macro that was customized in the installation steps.

7. Open the SAMPJCL member SCM1HOLD in an edit session and execute the SCMSEDIT macro from the command line.

   Note: Update SCM1HOLD SAMPJCL to download the HOLDDATA file.

   SCM1HOLD is customized.

8. Submit SCM1HOLD.

   The job downloads the external HOLDDATA file.

9. Open the SAMPJCL member SCM7RECH in an edit session and execute the SCMSEDIT macro from the command line.

   SCM7RECH is customized.

10. Submit SCM7RECH.

    The job receives the external HOLDDATA file.

11. (CA Recommended Service (CA RS)) installation only) Do the following:
    a. Determine which ASSIGN statements to download.
        - The yearly CA RS ASSIGN statements are stored in the following file:

          `ftp.ca.com/pub/ASSIGN/YEARLY/CAR`*yyyy*`.TXT`
        - The quarterly CA RS ASSIGN statements are stored in the following file:

          `ftp.ca.com/pub/ASSIGN/CAR`*yymm*`.TXT`

    b. Open the SAMPJCL member SCM7CARS in an edit session, update SCM7CARS SAMPJCL to download ASSIGN statements from http://ca.com/support, and execute the SCMSEDIT macro from the command line.

    SCM7CARS is customized.
12. (CA RS installation only) Submit SCM7CARS.

    The job downloads the CA RS ASSIGN statements.
13. (CA RS installation only) Open the SAMPJCL member SCM7RECP in an edit session, manually add the data set that contains the ASSIGN statements to the SMPPTFIN DD, and execute the SCMSEDIT macro from the command line.

    SCM7RECP is customized.
14. (CA RS installation only) Submit SCM7RECP.

    The job receives the external HOLDDATA file and CA RS ASSIGN statements.
15. Open the SAMPJCL member SCM8APYP in an edit session and execute the SCMSEDIT macro from the command line.

    SCM8APYP is customized.
16. Submit SCM8APYP.

    The PTFs are applied.
17. (Optional) Open the SAMPJCL member SCM9ACCP in an edit session and execute the SCMSEDIT macro from the command line.

    SCM9ACCP is customized.
18. (Optional) Submit SCM9ACCP.

    The PTFs are accepted.

    Note: You do not have to submit the job at this time. You can accept the PTFs according to your site policy.

HOLDDATA

When you apply maintenance, you typically encounter SMP/E HOLDDATA. We use HOLDDATA to notify your SMP/E system of SYSMODs that have errors or special conditions. We support system and external HOLDDATA.

System HOLDDATA

System HOLDDATA indicates data that is an in-stream part of the SYSMOD, informing you of special conditions. The following reasons are used with SYSTEM HOLDDATA for your product:

ACTION

Indicates that you must perform special processing before or after you apply this SYSMOD.

AO

Affects automated operations. It changes either the message identifier or the displacement of a field inside the message.

DB2BIND

Indicates that DBRMs have changed and packages need to be rebound.

DDDEF

Indicates that data sets and DDDEFs are being added or modified.

DELETE

Deletes the SYSMOD load module. You cannot reverse this type of SYSMOD with the SMP/E RESTORE command.

DEP

Indicates a dependency for this SYSMOD that you must externally verify.

DOC

Indicates a documentation change with this SYSMOD.

DYNACT

Describes the steps to dynamically activate this fix without performing an IPL.

EC

Indicates that this SYSMOD requires a hardware engineering change. An EC hold SYSMOD usually does not affect the product unless the EC is present on the hardware device.

ENH

Introduces a small programming enhancement. The hold contains the instructions to implement the enhancement. If no action is needed to implement the enhancement, give a summary of the enhancement.

EXIT

Indicates that changes delivered by this SYSMOD require reassembly of user exits.

EXRF

Indicates that the SYSMOD must be installed in both the Active and Alternate Extended Recovery Facility Systems.

IPL

Indicates that an IPL is required for this SYSMOD to take effect. This is used only when there is no alternative for dynamic activation.

MSGSKEL

Indicates that the SYSMOD contains internationalized message versions that must be run through the message compiler for each language.

MULTSYS

Apply this SYSMOD to multiple systems for either pre-conditioning, coexistence, or exploitation.

RESTART

Indicates that after applying this SYSMOD, the site must perform a special restart as opposed to a routine restart.

SQLBIND

Indicates that a bind is required for a database system other than DB2.

DOWNLD

Indicates that some or all of the elements that this SYSMOD delivers are to be downloaded to a workstation.

Code a BYPASS(HOLDSYS) operand on your APPLY command to install SYSMODs that have internal holds. Code the BYPASS(HOLDSYS) operand only after you have performed the required action, or if you are performing the action after the APPLY, if that is appropriate.

External HOLDDATA

External HOLDDATA is not part of the PTF. The HOLDDATA resides in a separate file and contains both error and FIXCAT HOLDDATA. The error HOLDDATA is used for SYSMODs that have been distributed and later are discovered to cause problems. The FIXCAT HOLDDATA helps identify maintenance that is required to support a particular hardware device, software, or function.

Download the external HOLDDATA from http://ca.com/support to a DASD file, and allocate the file to the SMPHOLD DD statement. To take care of the external HOLDDATA, receive it into your SMP/E environment. SMP/E receives the HOLDDATA from CA-supplied jobs.

You can find JCL to download the external HOLDDATA in your SAMPJCL member. Open SCM1HOLD in an edit session and execute the SCMSEDIT macro on the command line. Then, submit the JCL.

Error HOLDDATA

If a SYSMOD has unresolved error HOLDDATA, SMP/E does not install it unless you add a bypass to your APPLY command. You can bypass error HOLDDATA in situations that are not applicable to you. Error HOLDDATA that is not applicable to you can include a problem that happens only with a hardware device that you do not have or in a product feature that you do not use.

When CA Technologies publishes a SYSMOD that resolves the error HOLDDATA, the resolving SYSMOD supersedes the error HOLDDATA. This action lets you apply the original SYSMOD in conjunction with the fixing SYSMOD.

The only manual task is running a REPORT ERRSYSMODS. This report identifies the following:

- Any held SYSMODs already applied to your system
- Any resolving SYSMODs that are in RECEIVE status

SMP/E identifies the SYSMOD to apply to correct the situation.

FIXCAT HOLDDATA

CA Technologies provides FIXCAT HOLDDATA to help identify maintenance that is required to support a particular hardware device, software, or function. Fix categories are supplied as SMP/E FIXCAT HOLDDATA statements. Each FIXCAT HOLDDATA statement associates an APAR and its related fixing PTF to one or more fix categories.

Chapter 5: Starting Your Product

This section describes what you need to do to start CA Endevor SCM.

This section contains the following topics:

Introduction

After you have installed the product files using CA CSM or ESD to an SMP/E CSI, additional tasks are required. These tasks prepare the SMP/E target files and your installation environment to enable you to configure CA Endevor SCM as appropriate for your organization. The necessary tasks are referred to as deployment and configuration tasks. How you perform these tasks depends on whether you are using CA CSM. How you perform these tasks also depends on whether you have an existing installation of CA Endevor SCM or you are installing CA Endevor SCM for the first time.

Note: During deployment, you copy the installed SMP/E target libraries to deployment libraries. When you configure deployment libraries, you create what are referred to as runtime libraries.

How to Deploy With CA CSM

After the product files are installed, you can create a deployment using CA CSM. To create a deployment, click the Create Deployment tab in the SMP/E Environment for the installed instance of CA Endevor SCM. Then, follow the instructions in CA CSM.

How to Deploy Without CA CSM

To create a deployment without using CA CSM, create copies of the SMP/E target libraries using standard IBM utilities. These copies are your deployment libraries. Use these libraries for your configuration.

How to Configure for Existing Clients

After the product files are deployed, the following tasks are required if you have an existing installation of CA Endevor SCM.

1. **Configure the deployed files.** The configuration tasks prepare the new release of CA Endevor SCM to be started and customized. For more information, see How to Complete Configuration Without CA CSM (see page 62) and How to Configure the Deployed Product Data Sets (see page 63).
2. **Verify the installation environment**. Verify that your installation environment is appropriately configured. Some or all of these tasks may have been completed when an earlier release of CA Endevor SCM was installed. For more information, see How to Complete Configuration Without CA CSM (see page 62) and How to Configure the Installation Environment (see page 72).
3. **Test the new release of CA Endevor SCM.** If you are an existing client and CA Endevor SCM is already installed and configured at your site, you can use the Sample Application. Alternatively, you can use your existing data to test your installation of the new release of CA Endevor SCM. For more information, see the chapter "Implementing the Sample Application (see page 83)" or the chapter "Testing New Releases with Your Data (see page 107)."

Note: The Software Configuration Component (SCS) of CA CSM is not recommended for use by existing clients, even if you installed using CA CSM. Therefore, configuring the installed files and verifying the installation environment (steps 1 and 2 in this topic) are described in How to Complete Configuration *Without* CA CSM.

How to Configure for New Clients

After the product files are deployed, the following tasks are required if you are installing CA Endevor SCM for the first time. The method you use to complete these tasks depends on how you installed the files.

- If you deployed the product files using CA CSM and are installing CA Endevor SCM for the first time, you can use the Software Configuration Service (SCS). SCS helps you configure and customize the files using the Rapid Implementation method to set up the software lifecycle and inventory structure. For more information, see How to Complete Configuration with CA CSM.

  Note: The CA CSM configuration process implements a Rapid Implementation configuration for CA Endevor SCM. Instead of using CA CSM to configure CA Endevor SCM, we recommend that you use the Best Practice Implementation. For more information about the Best Practice Implementation, see the *Scenario Guide*.

- If you did *not* use CA CSM to deploy the files, you must complete the configuration tasks without using SCS. For more information, see How to Complete Configuration Without CA CSM (see page 62).

Regardless of the installation method you must now do the following:

1. **Configure the deployed files.** The configuration tasks prepare CA Endevor SCM to be started and customized.
2. **Configure the installation environment**. The installation environment must be configured to support CA Endevor SCM.
3. **Customize CA Endevor SCM.** You must customize the product to define the software lifecycle, inventory structure, and to set options using one of the following methods.

   Note: The CA CSM configuration process implements a Rapid Implementation configuration for CA Endevor SCM. Instead of using CA CSM to configure CA Endevor SCM, we recommend that you use the Best Practice Implementation. For more information about the Best Practice Implementation, see the *Scenario Guide*.

   a. We recommend the Best Practice Implementation method. However, you can use the Rapid Implementation method to customize CA Endevor SCM.

      Note: The SCS component of CA CSM leads you through the steps required to set up a Rapid Implementation.

   b. You can use the Sample Application to test your implementation of CA Endevor SCM. The Sample Application can help you gain a better understanding of the features CA Endevor SCM provides. Then, you could use the Best Practice Implementation method. Instead of using Best Practice Implementation, you could design and define your lifecycle, inventory structure and options using the foreground ISPF panels or batch administration. For more information, see the chapter "Implementing the Sample Application (see page 83)" and the Best Practice Implementation in the *Scenario Guide*.

How to Complete Configuration With CA CSM

Note: The CA CSM configuration process implements a Rapid Implementation configuration for CA Endevor SCM. Instead of using CA CSM to configure CA Endevor SCM, we recommend that you use the Best Practice Implementation. For more information about the Best Practice Implementation, see the *Scenario Guide*.

If you used CA CSM to install and deploy the product files and you are installing CA Endevor SCM for the first time, you can use the Software Configuration Service (SCS). SCS automates the configuration of CA products after they have been installed and deployed using CA CSM.

The SCS steps include the tasks required to configure the deployed files to enable CA Endevor SCM to be started and customized. Also, the SCS automated configuration for CA Endevor SCM leads you through all the tasks necessary to perform a Rapid Implementation. The Rapid Implementation steps set up a fully scalable software development environment that is complete and includes a lifecycle and inventory structure. If this is the first time CA Endevor SCM is being installed at your site, we recommend that you use SCS to configure a Rapid Implementation.

SCS guides you through the steps necessary to configure CA Endevor SCM. All these steps are listed and described in the online SCS procedure. However, some steps are performed outside of SCS. The steps performed outside of SCS are referred to as the manual steps. These manual steps enable you to perform the following tasks:

- Configuration tasks required to prepare CA Endevor SCM for customization. These tasks include configuring the deployed product files and your installation environment. These tasks are required whether you installed the product files using CA CSM or EDS.
- Customization tasks required to set up a Rapid Implementation configuration.

A list of the manual steps follows. Cross references are provided to indicate where you can find more information about how to perform each step.

The following manual steps prepare the deployed files or your installation environment:

1. Availability of the Include Members.

   For more information, see How to Make Include Members SCMM@LIB and SCMM@SYM Available (see page 69).

2. Authorize the CSIQAUTU or CSIQAUTH libraries or copy the content to an existing authorized library.

   For more information, see How to Authorize CSIQAUTU and CSIQAUTH Libraries (see page 69).

3. LINKLIST/LPA Considerations Part 1.

   For more information, see How to Authorize CSIQAUTU and CSIQAUTH Libraries (see page 69).

4. LINKLIST/LPA Considerations Part 2

   For more information, see How to Authorize CSIQAUTU and CSIQAUTH Libraries (see page 69).

5. Define CA Endevor SCM to the ISPF environment.

   For more information, see How to Define CA Endevor SCM to the ISPF Environment (see page 75).

6. Define CA Endevor SCM to GRS (if not already done).

   For more information, see Global Resource Sharing (see page 72).

7. Modify the TSO Command Limiting Facility to allow ISPTLIB commands (ACMQ, BC1PACMI, ESORT, EM, OM, etc) - CA ACF2 for z/OS users only.

   For more information, see ISP Command Table Considerations (see page 31).

8. Establish security for CA Endevor SCM data sets.

   For more information, see How to Secure Your Data Sets (see page 80).

9. Secure Access to functions.

   For more information, see How to Establish Functional Security (see page 81).

Note: The following SCS step is executed after you complete the steps previously listed. However, it is not a manual step: Execute the @@UPDJCL SCMUPDTE utility program.

The following manual steps set up a Rapid Implementation.

Note: For more information about steps 1 through 20, see How to Set Up the Standard Environment (see page 141) in the appendix "Using Rapid Implementation."

1. Define the EA System files.
2. Process the CA Endevor SCM Tables.
3. Define the ACMQ files.
4. Define the Package file.
5. Define the Element Catalog and Index files.
6. Define the Master Control files.
7. Define the DEV, QA, and PRD APPL@SYS files.
8. Define the DEV, QA, and PRD COMM@SYS files.
9. Define the EA System.
10. Add Processors to the EA System.
11. Define the EA System Types.
12. Add the CA Endevor SCM Tables to the EA System.

13. Promote the CA Endevor SCM Tables and Processors.
14. Copy the Table Load Modules to the CSIQAUTU library.
15. Define the APPL@SYS System in the PRD Environment.
16. Define the APPL@SYS System in the QA Environment.
17. Define the APPL@SYS System in the DEV Environment.
18. Define the COMM@SYS System in the PRD Environment.
19. Define the COMM@SYS System in the QA Environment.
20. Define the COMM@SYS System in the DEV Environment.
21. Verify the configuration is complete.

    For more information, see How to Verify the Implementation (see page 155) in the appendix "Using Rapid Implementation."

22. Optional - Jobs to Delete Rapid Implementation files.

    For more information, see Optional Jobs to Delete Rapid Implementation Files (see page 154) in the appendix "Using Rapid Implementation."

After you have completed these manual steps, the lifecycle and inventory structure are set up and verified, you can begin to load your own inventory. For more information, see How to Complete Your Basic Implementation (see page 159) in the appendix "Using Rapid Implementation."

How to Complete Configuration Without CA CSM

The tasks in this section must be completed whether you installed the product files using CA CSM or ESD. If you used CA CSM, these tasks are required regardless of whether you are using SCS. The configuration tasks described in this section are listed and described in the online SCS steps. However, these tasks are not automated in SCS and must be performed manually outside of SCS.

To configure the installed files to enable CA Endevor SCM to be started and customized, complete the following configuration tasks:

1. Configure the deployed product data sets (see page 63).
   a. Modify deployed file names to your user-defined values using job BC1JJB03. For more information, see Edit the CSIQJCL Data Set (see page 64).
   b. Make SCMM@LIB and SCMM@SYM available to jobs at runtime (see page 69).

c. Authorize the CSIQAUTU or CSIQAUTH libraries (see page 69) or copy the content to an existing authorized library.

Note: A client with an existing installation may want to run their current release and the new release on the same system/LPAR. In this case, the client must verify that the libraries for the new release cannot be used by accident. For more information, see Concurrent Releases.

2. Configure the installation environment (see page 72)

Note: Existing clients could have already completed these steps for an earlier release of CA Endevor SCM.

a. Define CA Endevor SCM to GRS (see page 72).

b. If your site is using CA ACF2 security, modify the TSO Command Limiting Facility (see page 31) to allow ISPTLIB commands. These commands include ACMQ, BC1PACMI, ESORT, EONLY, EM, QM, and so on.

c. Create a CA Endevor SCM environment for the new release under TSO. For more information, see How to Define CA Endevor SCM to the ISPF Environment (see page 75).

d. Set up security as follows:

- Secure your data sets (see page 80) using the Alternate ID.
- Secure access to functions (see page 81).

How to Configure the Deployed Product Data Sets

The deployed product data sets must be configured to enable you to start and customize the product.

To prepare the deployed files, complete the following steps:

1. To edit the members of the *iprfx.iqual*.CSIQJCL data set, you customize the BC1JJB03 member to the naming conventions in effect at your site. Then submit the job for execution to perform global change commands against several of the deployed source libraries.

Note: The BC1JJB03 job performs global edits against some of the SMP/E runtime (deployed) libraries. BC1JJB03 uses the CA Endevor SCM program SCMUPDTE to make these edits. The SCMUPDTE program can run in report or update mode. For more information about SCMUPDTE, see SCMUPDTE Program (see page 68).

2. Make *iprfx.iqual*.CSIQJCL data set members SCMM@LIB and SCMM@SYM available to CA Endevor SCM jobs at runtime.
3. Authorize the load module libraries.

Note: A client with an existing installation may want to run their current release and the new release on the same system/LPAR. In this case, the client must verify that the libraries for the new release cannot be used by accident. For more information, see Concurrent Releases.

Edit the CSIQJCL Data Set

To customize the data set to the naming conventions in effect at your site, all users must edit BC1JJB03 to update some variables. If you plan to implement the Sample Application, additional variables must be edited in BC1JJB03.

Edit the CSIQJCL data set

1. Verify that the execution mode for SCMUPDTE is set to your preference. The BC1JJB03 job performs global edits against some of the SMP/E runtime (deployed) libraries. These libraries are created by the SMP/E installation process. The SCMUPDTE program can run in report or update mode. For more information about SCMUPDTE, see SCMUPDTE Program (see page 68).

   The update mode for the edits on the runtime libraries is set to your preference.
2. Move BC1JJB03 from *iprfx.iqual*.CSIQJCL to a separate library.

   Important: Running BC1JJB03 updates the *iprfx.iqual*.CSIQJCL library, and you do not want to update the job itself. So, BC1JJB03 must be moved from *iprfx.iqual*.CSIQJCL before you edit this file.

3. Change the following variables in BC1JJB03 to reflect the naming conventions in effect at your site. These changes are required for all users. After you make the appropriate changes, save the updated member.

   Note: Change only the values to the right of the < on the data input lines. Do not change values to the left of the < in delivered members.

   mycart

   Specifies the unit name for the cartridge device you are using.

   mydvolser

   Specifies the volume serial number of the disk used to store permanent data sets.

   - If your site does *not* require the use of the VOL=SER parameter when storing permanent data sets, do the following. Verify that the two lines in BC1JJB03 that eliminate this parameter are enabled. The product is delivered with this setting.
   - If your site does require the VOL=SER parameter, delete the two lines that eliminate this DVOLSER. Also, enable the line that changes DVOLSER to MYDVOLSER, providing the DVOLSER value for your site.

   myiprfx

   Specifies the highest-level qualifier used when assigning data set names for installation and execution libraries.

   myiqual

   Specifies the second-level qualifier used when assigning data set names for installation and execution libraries.

   mypdisk

   Specifies the unit label for permanent disk data sets.

   mytdisk

   Specifies the unit label for temporary disk data sets.

   myuprfx

   Specifies the highest-level qualifier used when assigning data set names for CA Endevor SCM user files.

   myuqual

   Specifies the second-level qualifier used when assigning data set names for CA Endevor SCM user files.

myvvolser

Specifies the volume serial number of the disk on which the VSAM data sets will be allocated.

- If your site does not require the use of the VOLUMES parameter when allocating VSAM data sets, do the following. Verify that the line in BC1JJB03 that comments this parameter out is enabled. The product is delivered with this setting.
- If your site does require this parameter, delete the line that comments out the VOLUMES parameter. Also, enable the line that changes VVOLSER to MYVVOLSER, providing the VVOLSER value for your site.

The BC1JJB03 member is customized to the naming conventions in effect at your site.

4. (Optional) Change the following variables, if you are implementing the Sample Application. After you make the changes appropriate for your site, save the updated member. Default values are supplied for all of these variables.

   Note: For more information about installing and using the Sample Application, see The Sample Application (see page 83).

   ADPROC?

   Indicates whether processors are in use at your site.

   - If your site received an LMP key for product A9, Extended Processors, enable the line that sets ADPROC? to ADPROC. The product is delivered with this setting.
   - If your site did *not* receive an LMP key, delete the line that sets this value to ADPROC and enable the line that sets ADPROC? to ADNOPROC.

   MONITOR=COMPONENTS

   Indicates that the CA Endevor Automated Configuration option is in use at your site. The product is delivered with this setting.

   If your site did *not* receive an LMP key for the CA Endevor Automated Configuration option, enable the line that changes MONITOR=COMPONENTS to MONITOR=NONE.

   SYSMACLIB

   Specifies the data set name of the system assembler macro library. As delivered, this value is set to SYS1.MACLIB.

   The BC1JJB03 member is customized for your site to use the Sample Application.

5. (Optional) Change the following variables as appropriate for your site, if you are using the Sample Application. BC1JJB03 uses the SCMUPDTE syntax to change these values (left of the <) to your preference (right of the <). For more information about SCMUPDTE, see SCMUPDTE Program Consideration (see page 68).

   Note: Your site can be using one or more types of extended processors. The Sample Application installs one COBOL and assembler processor.

   C??

   Specifies the version of COBOL in use at your site, either COBOL II or COBOL/LE.

   - If you are using COBOL/LE, change this value to CLE.
   - If you are using COBOL II, change this value to CII.
   - If your site uses both COBOL/LE and COBOL II, change the value to the version of COBOL you use most often.
   - CLE is the delivered default.

   C??NBL

   Specifies the default processor group for COBOL elements delivered with the Sample Application. The value is CIINBL or CLENBL, with the C?? part of this variable set to the same value as the variable C??. CLENBL is the delivered default.

   SYSCLECOMP

   Specifies the COBOL/LE compiler load library. IGY.SIGYCOMP is the delivered default.

   SYSCLERUN

   Specifies the z/OS Language Environment runtime library. CEE.SCEERUN is the delivered default.

   SYSCLELKED

   Specifies the z/OS Language Environment Automatic Call (link) library. CEE.SCEELKED is the delivered default.

   SYSCIICOMP

   Specifies the COBOL II compiler load library. SYS1.COB2COMP is the delivered default.

   SYSCIILIB

   Specifies the COBOL II Automatic Call (link) library. SYS1.COB2LIB is the delivered default.

   The BC1JJB03 member is customized for your site to use the Sample Application.

6. Submit job BC1JJB03 for execution with a valid JOB statement.

   The CSIQJCL data set is updated to change delivered default values to the values you specified as appropriate for your site.

SCMUPDTE Program

BC1JJBO3 uses the SCMUPDTE program to perform global edits on some of the SMP/E target libraries. SCMUPDTE executes in one of the following modes:

- **Non-update execution mode**–When the update execution parameter is not used, the PDSE members are not updated. However, all reports indicate what would have occurred during an update execution mode.
- **Update execution mode**–When the update execution parameter is specified, string 1 is replaced with string2 in all members of the input PDSEs.

  During execution mode, SCMUPDTE performs edits on the PDSEs as follows:

  - String1:
    - String1 is all of the characters up to the first less than sign (<) of the input (sysin) parameter.
    - String1 must be fully contained in positions 1 thru 71 of the input record and must be a "word" to be eligible for replacement.
    - String1 is considered to be a word, if it is both preceded and followed by any of the following characters:

      ' , . / < > ? " : ; ^ + _ ( ) * % | = and blank.
  - String2:
    - String2 is all of the characters after the first less than sign (<) and up to the second less than sign.
    - String2 can be of equal, greater, or less length than string1 and can be null.
  - Any characters after the second < are treated as line comments and are ignored.
  - If columns 1 through 3 contain the characters </*, the entire line is treated as a comment.
  - The maximum number of input parameters that can be specified is 40 (comment lines are not counted).
  - Syntax:
    - The maximum number of combined characters for string1 and string2 is 70.
    - The input parameter statements cannot be continued.
    - Data in columns 72-80 is ignored and unchanged
  - Supported input library characteristics:

    lrecl=80,recfm=f(b) and the first character of the ddname(s) must be "@"

How to Make Include Members SCMM@LIB and SCMM@SYM Available

Job BC1JJB03 edits members in the *iprfx.iqual.*CSIQJCL library that get included in other CA Endevor SCM job streams at runtime. The following members must be made available to CA Endevor SCM jobs at runtime:

- SCMM@LIB contains the runtime STEPLIB/CONLIB DD statements.
- SCMM@SYM contains a list of JCL SET statements defining symbolic parameter values.

To do this, use one of the following methods:

- Copy both of these members from the *iprfx.iqual.*CSIQJCL library to a JES PROCLIB library.
- If you do not want to move these members to a JES PROCLIB library, you can use an alternate method. To use the alternate method, add a JCLLIB statement in each job stream before executing it. The JCLLIB statement can be placed following the JOB statements. One or more libraries can be identified on this statement. An example JCLLIB statement follows:

```
//ENDVR02A JOB (41200000),'DUMMY JOBCARD',
//   REGION=4096K,CLASS=A,MSGLEVEL=(1,1),
//   MSGCLASS=X
// JCLLIB ORDER=(iprfx.iqual.CSIQJCL,uprfx.uqual.INCLUDE)
//-------------------------------------------------------
```

Note: Two members named SCMM@LIB and SCMM@SYM also exist in the *iprfx.iqual.*CSIQSENU runtime library. These two members contain the same information as the members in the JCLLIB library. However, the format of the data is a little different, because they are included in ISPF skeleton members using the IM (Include Member) command. These members must remain in the *iprfx.iqual.*CSIQSENU runtime library.

How to Authorize CSIQAUTU and CSIQAUTH Libraries

Note: The *iprfx.iqual*.CSIQAUTH and *iprfx.iqual*.CSIQAUTU data sets are created by SCM1ALL and populated by SCM3RECT, SCM4APP, and SCM5ACCT.

The load modules contained in *iprfx.iqual*.CSIQAUTH and *iprfx.iqual*.CSIQAUTU must reside in a system-authorized library. Use one of the following methods to authorize the libraries:

- Copy the members from the *iprfx.iqual*.CSIQAUTH and *iprfx.iqual*.CSIQAUTU libraries to an existing authorized library.
- Authorize the *iprfx.iqual*.CSIQAUTH and *iprfx.iqual*.CSIQAUTU libraries.
- Copy the members from *iprfx.iqual*.CSIQAUTH and *prfx.iqual*.CSIQAUTU (and optionally *iprfx.iqual*.CSIQLOAD) to an existing LINKLIST defined library. This method assumes that the LINKLIST data set is authorized.

- Copy the members from *iprfx.iqual*.CSIQAUTH and *iprfx.iqual*.CSIQAUTU (and optionally *iprfx.iqual*.CSIQLOAD) to an existing LPA defined library. LPA libraries are authorized. If you use this method, then perform the following additional tasks:
  - Before moving a library to an existing LPA library, inspect the members and move those members that are non-reentrant to a LINKLIST-defined library. Only reentrant load modules can be executed from the LPA. For more information, see How to Determine Which Members are Non-Reentrant (see page 70).
  - You may want to determine the storage requirements for LPA-eligible load modules. For more information, see How to Determine LPA Load Module Storage Requirements (see page 71).
  - Prepare certain data set members. For more information, see How to Prepare Members in LINKLIST or LPA Libraries (see page 71).
- Define the *iprfx.iqual*.CSIQAUTH and *iprfx.iqual*.CSIQAUTU (and optionally *iprfx.iqual*.CSIQLOAD) data sets as LINKLIST or LPA libraries themselves. If you use this method, then perform the following additional tasks:
  - Before defining a library as an LPA library, inspect the members and move those members that are non-reentrant to a LINKLIST-defined library. Only reentrant load modules can be executed from the LPA. For more information, see How to Determine Which Members are Non-Reentrant (see page 70).
  - You may want to determine the storage requirements for LPA-eligible load modules. For more information, see How to Determine LPA Load Module Storage Requirements (see page 70).
  - Prepare certain data set members. For more information, see How to Prepare Members in LINKLIST or LPA Libraries (see page 71).

Note: If a LINKLIST library is used, it is likely that the data set is authorized. If an LPA library is used, it is authorized.

How to Determine Which Members are Non-Reentrant

Only reentrant load modules can be executed from the LPA. Before copying the contents of a library to an LPA defined library or defining a library as an LPA library, inspect the members to determine which ones are non-reentrant. Any non-reentrant members must be moved to a LINKLIST-defined library. To determine which members are non-reentrant, use the ISPF Browse facility and follow these steps:

1. Display an ISPF browse member selection list for the load library.
2. Issue the SORT ATTR command.
3. Scroll to the right to see the Attributes column.

   Note: Most members show attributes of RN RU, for reentrant and reusable.
4. Members showing spaces instead of RN are non-reentrant and cannot be moved into the LPA.

How to Determine LPA Load Module Storage Requirements

To determine storage requirements for LPA-eligible load modules, use the ISPF Browse facility and follow these steps:

1. Display an ISPF browse member selection list for the load library.
2. Issue the SORT ATTR command.
3. Scroll to the right to see the Attributes column.
4. Eliminate all programs where the reentrant attribute is blank, because these programs cannot be placed into the LPA.
5. Issue the SORT RMODE command.

   To determine the amount of storage required, follow one of these methods:

   - For 24-bit storage, add the hexadecimal value found in the Size field for all programs containing the value 24 in the RM (RMODE) field.
   - For 31-bit storage, add the hexadecimal value found in the Size field for all programs containing the value ANY in the RM (RMODE) field.

How to Prepare Members in LINKLIST or LPA Libraries

When LINKLIST or LPA libraries are used to authorize your data sets, certain member must be edited. Edit the data set members as follows:

- If you use LINKLIST or LPA libraries for the CSIQAUTH and CSIQAUTU data sets, do the following:
  - Remove the STEPLIB DD statements from the CA Endevor SCM sample job streams.
  - Remove, or comment out, the STEPLIB DD statements in the SCMM@LIB members in the CSIQJCL and CSIQSENU.
- If you use the LINKLIST and LPA libraries for the *iprfx.iqual*.CSIQLOAD data set, do the following:
  - Remove the CONLIB DD statement from the sample job streams.
  - Remove, or comment out, the CONLIB DD statements in the SCMM@LIB members in the CSIQJCL and CSIQSENU libraries.

How to Configure the Installation Environment

The installation environment must be configured to support your installation of CA Endevor SCM. To do this, complete the following tasks:

- If your site has multiple CPUs sharing DASD, you must define queue names for CA Endevor SCM. For more information, see Global Resource Sharing (see page 72).
- Create a CA Endevor SCM environment for the new release under TSO to enable users and administrators access to the product. For more information, see How to Define CA Endevor SCM to the ISPF Environment (see page 75).
- Set up security as follows:
  - Secure your data sets using the Alternate ID.
  - Secure access to functions.
- If your site is using CA ACF2 security, modify the TSO Command Limiting Facility (see page 31) to allow ISPTLIB commands. These commands include ACMQ, BC1PACMI, ESORT, EONLY, EM, QM, and so on.

Global Resource Sharing

If your site has multiple CPUs sharing DASD, you must define queue names. If you are using Unicenter CA-MIM Resource Sharing or IBM Global Resource Serialization (GRS), include the queue names in the appropriate Global Resource Queue Name Table.

How to Define Queue Names for Unicenter CA-MII ENQ Management Definitions

The Unicenter CA-MII component of Unicenter CA-MIM has two operating modes, ALLSYSTEMS mode and SELECT mode. If you are using Unicenter CA-MIM Resource Sharing, the ENQ management definitions for CA Endevor SCM must be added in the Unicenter CA-MII MIMQNAME parameter library member. Which definitions depend on which mode Unicenter CA-MII is operating.

To define the queue names, do the following:

1. Determine which operating mode your active Unicenter CA-MII address space is running in. To do this, issue the Unicenter CA-MIM command “F MIM,D INIT” and review the results. To determine if PROCESS=ALLSYSTEMS or PROCESS=SELECT is in effect, review the MIM1019I command response message.

2. Add the definitions depending on which mode is running, as follows:

   - ALLSYSTEMS processing mode—If you are running Unicenter CA-MII in ALLSYSTEMS mode, add the following ENQ management definitions for CA Endevor SCM in the Unicenter CA-MII MIMQNAME parameter library member.

```
CTLIELEM GDIF=YES,
         SCOPE=SYSTEMS,
         EXEMPT=NO,
         ECMF=YES,
         RPTAFTER=30,
         RPTCYCLE=60

CTLIPROC GDIF=YES,
         SCOPE=SYSTEMS,
         EXEMPT=NO,
         ECMF=YES,
         RPTAFTER=30,
         RPTCYCLE=60

ENDEVOR  GDIF=YES,
         SCOPE=SYSTEMS,
         EXEMPT=NO,
         ECMF=YES,
         RPTAFTER=30,
         RPTCYCLE=60

SPFEDIT  GDIF=YES,
         SCOPE=SYSTEMS,
         EXEMPT=YES,
         ECMF=YES,
         RPTAFTER=0,
         RPTCYCLE=60
SYSIEWLP GDIF=YES,
         SCOPE=SYSTEMS,
         EXEMPT=NO,
         ECMF=YES,
         RPTAFTER=30,
         RPTCYCLE=60
```

- SELECT processing mode—If you are running Unicenter CA-MII in SELECT mode, add the following ENQ management definitions for CA Endevor SCM in the Unicenter CA-MII MIMQNAME parameter library member.

```
CTLIELEM GDIF=YES,
         SCOPE=SYSTEMS,
         EXEMPT=NO,
         ECMF=YES,
         RPTAFTER=30,
         RPTCYCLE=60

CTLIMSTR GDIF=YES,
         SCOPE=SYSTEMS,
         EXEMPT=NO,
         ECMF=NO

CTLIPROC GDIF=YES,
         SCOPE=SYSTEMS,
         EXEMPT=NO,
         ECMF=YES,
         RPTAFTER=30,
         RPTCYCLE=60

ENDEVOR  GDIF=YES,
         SCOPE=SYSTEMS,
         EXEMPT=NO,
         ECMF=YES,
         RPTAFTER=30,
         RPTCYCLE=60

SPFEDIT  GDIF=YES,
         SCOPE=SYSTEMS,
         EXEMPT=YES,
         ECMF=YES,
         RPTAFTER=0,
         RPTCYCLE=60
SYSIEWLP GDIF=YES,
         SCOPE=SYSTEMS,
         EXEMPT=NO,
         ECMF=YES,
         RPTAFTER=30,
         RPTCYCLE=60
```

Queue names are defined to allow CA Endevor SCM to support a shared DASD environment.

GRS Definitions

If you are using IBM Global Resource Serialization (GRS), the following queue name considerations apply:

- There should be no reason to have INCL statements for CA Endevor SCM QNAMEs. Using the CON will convert the RESERVE on the volume, and by default GRS will include the SCOPE=SYSTEMS enqueue for global propagation.
- Only the QNAMES of SPFEDIT, SYSIEWLP, ENDEVOR and CTLIMSTR are used with RESERVE W/SCOPE=SYSTEMS + UCB locks. The remaining QNAMES, CTLIPROC, and CTLIELEM are used with ENQUEUEW/SCOPE=SYSTEMS.

Sample definition for when your site decides to CONVERT RESERVEs:

```
/*************************************************************/
/* RESOURCE NAME LIST FOR CA ENDEVOR SCM */
/*************************************************************/
RNLDEF RNL(CON) Type(GENERIC) QNAME(CTLIMSTR)
RNLDEF RNL(CON) Type(GENERIC) QNAME(SPFEDIT)
RNLDEF RNL(CON) Type(GENERIC) QNAME(SYSIEWLP)
RNLDEF RNL(CON) Type(GENERIC) QNAME(ENDEVOR)
```

How to Define CA Endevor SCM to the ISPF Environment

CA Endevor SCM must be defined to the ISPF environment. You must identify the CA Endevor SCM data sets to ISPF, using either a CLIST or existing logon procedures.

Before you define CA Endevor SCM to the ISPF environment, consider the following information:

- Many factors influence the region size required to execute CA Endevor SCM. These factors include split-screen processing, the use of CA Panvalet or CA Librarian, the maximum size of elements, whether sort work areas are defined in the CLIST, and whether ISPF resides in the LPA. A region size of 4096KB is sufficient for most installations.
- The delivered CLIST library (*yourHLQ*.CSIQCLS0) is fixed block (FB). If your existing CLIST libraries are variable blocked (VB), use the TSO utilities to create a VB file.

To identify the data sets to ISPF, do the following:

1. Identify the files to ISPF using one of the following methods:
   - Use the ISPF LIBDEF Services. (We recommend this method.)
   - Use the TSO FREE and ALLOCATE CLIST services to reallocate the ISPF libraries.
   - Allocate all data sets through TSO logon procedures.

2. Define an ISPF option for CA Endevor SCM to the main ISPF dialog by adding CA Endevor SCM as a valid option on an ISPF primary or secondary options panel.
3. Execute the CLIST.

   The CLIST is executed automatically when you select the CA Endevor SCM option from the ISPF main dialog.

The ISPF LIBDEF Services Method

You can use the ISPF LIBDEF services to identify CA Endevor SCM libraries to the ISPF environment. Use member ENDEVOR provided in the installed CSIQCLS0 library. We recommend this approach.

Note: For the Japanese installation, you must edit the ENDEVOR member to include the Japanese panel and menu library names in the appropriate data set concatenations. For more information, see Edit ENDEVOR Member for Japanese Installation (see page 246).

Copy the CLIST to a data set that is allocated to the standard SYSPROC DD concatenation.

If the authorized CA Endevor SCM load modules have not been placed in LPA, LINKLIST, or STEPLIB libraries, you can use the ISPF ISPLLIB DD statement instead. However, allocation of an authorized library under ISPLLIB may not support all necessary system authorization requirements. Allocate ISPLLIB to the CA Endevor SCM authorized load libraries before invoking ISPF.

The CLIST you see at your site is an edited version, reflecting the information entered during the installation of the INSTALL job. After you create the CLIST, change an ISPF primary or secondary options panel to include an option for CA Endevor SCM.

Modify your ISPF panel similar to the following sample ISPF/PDF Primary Option Menu. The sample panel has been modified to add option E for CA Endevor SCM. When you select this option, the ENDEVOR CLIST is invoked to begin the CA Endevor SCM dialog.

The following lines, which are shown in bold in the sample, enable users to access CA Endevor SCM.

- **E +Endevor - Endevor Software Management System —** Indicates what the user enters to select CA Endevor SCM.
- **E,'CMD(Endevor)' —** Indicates that the ENDEVOR CLIST is to be executed when the defined option is selected.

```
%----------------------- ISPF/PDF PRIMARY OPTION MENU ----------------------
%OPTION ===>_ZCMD +
%
       +USERID - &ZUSER
% 0 +ISPF PARMS - Specify terminal and user parameters           +TIME - &ZTIME
% 1 +BROWSE - Display source data or output listings
       +TERMINAL - &ZTERM
% 2 +EDIT - Create or change source data                                 +PF
KEYS - &ZKEYS
% 3 +UTILITIES - Perform utility functions
% 4 +FOREGROUND - Invoke language Processors in foreground
% 5 +BATCH - Submit job for language processing
% 6 +COMMAND - Enter TSO Command, CLIST, or REXX exec
% 7 +DIALOG TEST - Perform dialog testing
% 8 +LM UTILITIES- Perform library administrator utility functions
% 9 +IBM PRODUCTS- Additional IBM program development products
% 10 +SCLM - Software Configuration and Library Manager
% E +Endevor - Endevor Software Management System
% C +CHANGES - Display summary of changes for this release
% T +TUTORIAL - Display information about ISPF/PDF
% X +EXIT - Terminate ISPF using log and list defaults
%
+Enter%End+command to terminate ISPF.
)INIT
)PROC
&ZQ = &Z
  IF (&ZCMD ¬= ' ')
    &ZQ = TRUNC(&ZCMD,'.')
    IF (&ZQ = ' ')
      .MSG = ISRU000
  &ZSEL = TRANS( &ZQ
                0,'PANEL(ISPOPTA)'
                1,'PGM(ISRBRO) PARM(ISRBRO01)'
                2,'PGM(ISREDIT) PARM(P,ISREDM01)'
                3,'PANEL(ISRUTIL)'
                4,'PANEL(ISRFPA)'
                5,'PGM(ISRJB1) PARM(ISRJPA) NOCHECK'
                6,'PGM(ISRPTC)'
                7,'PGM(ISPYXDR) PARM(ISR) NOCHECK'
                8,'PANEL(ISRLPRIM)'
                9,'PANEL(ISRDIIS)'
               10,'PGM(ISRSCLM) NOCHECK'
                E,'CMD(Endevor)'
                C,'PGM(ISPTUTOR) PARM(ISR00005)'
                T,'PGM(ISPTUTOR) PARM(ISR00000)'
              ' ',' '
                X,'EXIT'
                *,'?' )
  &ZTRAIL = .TRAIL
)End
```

The TSO FREE and ALLOCATE Method

You can use the TSO FREE and ALLOCATE services to identify CA Endevor SCM libraries to the ISPF environment. Use the CLIST provided on the installation cartridge, member BC1PCLS1 in the CSIQCLS0 library.

Note: For the Japanese installation, you must edit the BC1PCLS1 member to include the Japanese panel and menu library names in the appropriate data set concatenations. For more information, see Edit BC1PCLS1 Member for Japanese Installation (see page 247).

The CLIST must be copied to a data set in the standard SYSPROC DD concatenation and must be invoked at LOGON time.

The CLIST you see at your site is an edited version, reflecting the information entered during the installation of the INSTALL job.

After you define the CLIST, you must change an ISPF primary or secondary options panel to include an option for CA Endevor SCM. In the following sample, the ISPF/PDF Primary Option Menu has added option E for CA Endevor SCM. Whenever you select this option, CA Endevor SCM is automatically invoked. The first bold line indicates what to enter to select CA Endevor SCM. The second line indicates that the CA Endevor SCM program should be executed when the defined option is selected.

Modify your ISPF panel similar to the following sample panel.

```
%---------------------- ISPF/PDF PRIMARY OPTION MENU ----------------------
%OPTION ===>_ZCMD +
%
       +USERID - &ZUSER
% 0 +ISPF PARMS - Specify terminal and user parameters           +TIME - &ZTIME
% 1 +BROWSE - Display source data or output listings
       +TERMINAL - &ZTERM
% 2 +EDIT - Create or change source data                                 +PF
KEYS - &ZKEYS
% 3 +UTILITIES - Perform utility functions
% 4 +FOREGROUND - Invoke language Processors in foreground
% 5 +BATCH - Submit job for language processing
% 6 +COMMAND - Enter TSO Command, CLIST, or REXX exec
% 7 +DIALOG TEST - Perform dialog testing
% 8 +LM UTILITIES- Perform library administrator utility functions
% 9 +IBM PRODUCTS- Additional IBM program development products
% 10 +SCLM - Software Configuration and Library Manager
% E +Endevor - Endevor Software Management System
% C +CHANGES - Display summary of changes for this release
% T +TUTORIAL - Display information about ISPF/PDF
% X +EXIT - Terminate ISPF using log and list defaults
%
+Enter%End+command to terminate ISPF.
)INIT
)PROC
&ZQ = &Z
  IF (&ZCMD ¬= ' ')
    &ZQ = TRUNC(&ZCMD,'.')
    IF (&ZQ = ' ')
      .MSG = ISRU000
  &ZSEL = TRANS( &ZQ
                0,'PANEL(ISPOPTA)'
                1,'PGM(ISRBRO) PARM(ISRBRO01)'
                2,'PGM(ISREDIT) PARM(P,ISREDM01)'
                3,'PANEL(ISRUTIL)'
                4,'PANEL(ISRFPA)'
                5,'PGM(ISRJB1) PARM(ISRJPA) NOCHECK'
                6,'PGM(ISRPTC)'
                7,'PGM(ISPYXDR) PARM(ISR) NOCHECK'
                8,'PANEL(ISRLPRIM)'
                9,'PANEL(ISRDIIS)'
               10,'PGM(ISRSCLM) NOCHECK'
                E,'PGM(C1SM1000) NEWAPPL(CTL1) NOCHECK'
                C,'PGM(ISPTUTOR) PARM(ISR00005)'
                T,'PGM(ISPTUTOR) PARM(ISR00000)'
              ' ','' '
                X,'EXIT'
                *,'?' )
  &ZTRAIL = .TRAIL
)End
```

The TSO Logon Procedure Method

You can allocate the CA Endevor SCM data sets through your existing TSO logon procedures. Allocate the data sets as follows:

DDnames	Library Created by the INSTALL Jobs
SYSPROC	*iprfx.iqual*.CSIQCLS0
ISPPLIB	*iprfx.iqual*.CSIQPJPN (Japanese panels for installation of Japanese version) *iprfx.iqual*.CSIQPENU
ISPSLIB	*iprfx.iqual*.CSIQSENU
ISPMLIB	*iprfx.iqual*.CSIQMJPN (Japanese messages for installation of Japanese version) *iprfx.iqual*.CSIQMENU
CONLIB	*iprfx.iqual*.CSIQLOAD (if not in LINKLIST or LPA)
ISPTLIB	*iprfx.iqual*.CSIQTENU

After you copy these libraries to the logon procedure, change an ISPF primary or secondary options panel to include an option for CA Endevor SCM.

Note: For more information about how to add an option for CA Endevor SCM, see The TSO FREE and ALLOCATE Method.

How to Secure Your Data Sets

The CA Endevor SCM Alternate ID allows you to secure CA Endevor SCM data sets from updates outside of CA Endevor SCM while allowing users to perform functions that update those data sets from within CA Endevor SCM. Before any action that accesses a CA Endevor SCM data set, the Alternate ID is swapped for the user's ID and access is performed using the Alternate ID.

Important! We recommend that you use the Alternate ID to secure your data sets. For more information, see the chapter "Implementing Data Set Security" in the *Security Guide*. Typically, you secure a high level qualifier as READ/WRITE for CA Endevor SCM, READ/WRITE/UPDATE for the Administrator, and READ for all others.

To secure your CA Endevor SCM data sets, complete the following steps:

1. Define a user ID for CA Endevor SCM to use as an Alternate ID to your security software product (CA Top Secret, CA ACF2 or RACF).
2. Grant the Alternate ID CONTROL access to all CA Endevor SCM VSAM files and UPDATE access to Source Output and any other libraries and files that are to be written to.

   Note: A single HLQ (uprfx) is used in the Rapid Implementation setup job, so you can grant access to all the data sets using one rule. For example, this rule grants access to all the uprfx data sets: "uprfx.** R(A) W(A) C(A)"
3. Update the RACFUID= parameter in your C1DEFLTS table to supply the Alternate ID.

How to Establish Functional Security

The External Security Interface (ESI) allows you to establish security rules for access to your Environments, Systems, Subsystems, actions, and so on, within CA Endevor SCM. This is accomplished by defining pseudo data set names and functional access levels in the ESI table (BC1TNEQU) that are tested against the user's access level through the SAF (Security Authorization Facility) interface.

To secure access to functions, complete the following steps:

1. Secure access to functions by defining pseudo data set names and functional access levels in the External Security Interface (ESI) table (member BC1TNEQU).
2. The security settings are tested against the user's access level through the Security Authorization Facility (SAF) interface.

Note: For more information about functional security, see the *Security Guide*.

Chapter 6: Implementing the Sample Application

This section contains the following topics:

The Sample Application

Important! If you are installing CA Endevor SCM for the first time, we recommend that you use the Best Practice Implementation. For more information about the Best Practice Implemetation, see the *Scenario Guide*. If CA Endevor SCM is already installed at your site, you can use your existing inventory and lifecycle for testing purposes. For more information, see How to Test New Releases with Your Data (see page 107).

The CA Endevor SCM sample application is provided as a training tool to help you understand how to implement CA Endevor SCM. The sample application reflects a basic software lifecycle, including examples of a lifecycle definition, inventory definitions, and processors, all of which can be useful to you during your site's implementation of CA Endevor SCM.

The sample application presents one way of implementing CA Endevor SCM, which may be different from the way that you have implemented it at your site. It may include steps that you do not perform or it may present steps in a different order.

Before you implement the sample CA Endevor SCM application at your site, be sure that CA Endevor SCM has been installed successfully.

Sample Application Naming Conventions

The following is the naming conventions for the files used by the sample application.

`iprfx.iqual.SMPLssss.xxxxxxxx`

iprfx

Specifies the highest-level qualifier (1-8 characters in length) for the CA Endevor SCM and sample application libraries and files; for example, SYS3. This value was defined as part of the global edits performed prior to running the CA Endevor SCM installation process.

iqual

Specifies the second-level qualifier (1-8 characters in length) for the CA Endevor SCM and sample application libraries and files; for example, ENDEVOR.

SMPL*ssss*

Specifies the identifier used together with the stage name (indicated by the *ssss*) to construct the third qualifier as follows:

- TEST-For Stage 1 of SMPLTEST (unit test)
- QA-For Stage 2 of SMPLTEST (quality assurance)
- EMER-For Stage 1 of SMPLPROD (emergency fixes)
- PROD-For Stage 2 of SMPLPROD (production stage)

For example, if during the installation process you changed *iprfx* to SYS3 and *iqual* to ENDEVOR, your data sets would be named as follows:

- SYS3.ENDEVOR.SMPLTEST.*xxxxxxxx*
- SYS3.ENDEVOR.SMPLQA.*xxxxxxxx*
- SYS3.ENDEVOR.SMPLEMER.*xxxxxxxx*
- SYS3.ENDEVOR.SMPLPROD.*xxxxxxxx*

xxxxxxxx

Specifies the fourth-level qualifier of the data sets used for the sample application and represents the type of data contained in the file; for example, MCF for the master control file, PACKAGE for the package control file, and so on.

Important! File names are limited to a total of 44 characters, including periods. The maximum length of each data set name qualifier is eight characters.

How to Implement the Sample Application

To implement the sample application, perform the following basic tasks:

1. Edit the CA Endevor SCM Defaults Table (C1DEFLTS) (see page 85).
2. Define and allocate CA Endevor SCM libraries (see page 88).
3. Define the CA Endevor SCM inventory structure and populate it with an application (see page 93).
4. Verify the implementation (see page 94).
5. Perform ACM Query Facility verification. (see page 103)

All JCL and source members used in the implementation steps and the processors for the sample environments are included in the CA Endevor SCM *iprfx.iqual.CSIQSAMP* library created during the installation process. The members in this library were customized according to the values you supplied in installation job BC1JJB03.

Defaults Table (C1DEFLTS)

The CA Endevor SCM Defaults Table contains processing and environment definition information for your site. The table is comprised of a set of macros which, when assembled and link-edited, are known collectively as the C1DEFLTS Table. The types of C1DEFLTS macros are distinguished by the following TYPE keyword parameters:

- One TYPE=MAIN parameter is required for each site, to define general site-specific information. The TYPE=MAIN macro is the first macro in the definition of the table.
- Any number of TYPE=ENVRNMNT macros can be included in the table. You must have one TYPE=ENVRNMNT macro for each environment in your configuration. The TYPE=ENVRNMNT macros follow the TYPE=MAIN macro in the definition of the table.
- One TYPE=END macro is needed to indicate the end of the table definition. It follows the last TYPE=ENVRNMNT macro.

1-How to Edit C1DEFLTS

The sample application C1DEFLTS table is in the library iprfx.iqual.CSIQSAMP, in member SMPLDEFT. For the sample application, do the following:

1. Enter site-specific parameters in the TYPE=MAIN macro section of the C1DEFLTS table. Enter or edit the parameter values, as appropriate.

Important! Be sure the *iprfx.iqual* values you assign to data set names in the C1DEFLTS Table match those you defined as part of the global edits performed during installation. For more information, see The Variable Definitions Table (see page 113).

2. Verify MCF qualifiers in the TYPE=ENVRNMNT macro.

   For the purposes of the sample application, you cannot change anything in the TYPE=ENVRNMNT macro. Instead, verify that the qualifiers for the Master Control File (MCF) data sets match those specified during the global edits. The data set name is supplied during the definition of the sample application data sets.

   Note: For more information, see How to Define and Allocate CA Endevor SCM Libraries. Be sure to use the same data set names when allocating these data sets in job SMPLJOB2. For more information, see Run SMPLJOB2.

3. Run SMPLDEFT

   After you have edited the sample defaults table, you need to assemble and link-edit it. If you have not already done so, copy your JOBCARD member to the beginning of member SMPLDEFT in iprfx.iqual.CSIQSAMP, then submit the job for execution. Store the load module in your authorized library as member C1DEFLTS. After you have implemented and worked with the sample application and you are ready to build your own lifecycle, use member C1DEFLTS in *iprfx.iqual.CSIQSRC*.

The site-specific parameters in the TYPE=MAIN macro are as follows:

ACMROOT

(Required if you are using ACM) Names the data set name of the VSAM file your site will use to store the name of each CA Endevor SCM element and all its related components. This data set is required if your site is licensed to use ACM.

The data set name is supplied during the definition of the ACM Query data sets. For more information, see Allocating and Initializing ACM Query Files (SMPLACMD) (see page 91). Be sure and use this same data set name when allocating these data sets in job SMPLACMD.

If you are *not* using ACM, remove the data set name; for example: ACMROOT=**,**

Note: For more information about ACM, see the *Automated Configuration Option Guide*.

ACMXREF

(Required if you are using ACM) Names the data set name of the VSAM file you will use to store the name of each component relationship. This data set is required if your site is licensed to use ACM.

The data set name is supplied during the definition of the ACM Query data sets. For more information, see Allocating and Initializing ACM Query Files (SMPLACMD) (see page 91). Be sure and use this same data set name when allocating these data sets in job SMPLACMD.

If you are *not* using ACM, remove the data set name; for example: ACMXREF=**,**

Note: For more information about ACM, see the *Automated Configuration Option Guide*.

ASCM

Specifies a value, Y or N, to indicate whether your site is licensed for the Automated Configuration Option (ACM) facility.

If you received an LMP key for Product AY, ACM, enter Y. Otherwise, enter N.

CUNAME

Names the 1-50 character name that describes your site. This name is used in report headings.

ELMCATL

Names the 1-44 character data set name for the Element Catalog. The name is supplied during the definition of the Element Catalog. For more information, see Allocate the Master Control Files, Element Catalog, Base, Delta, and Output Libraries. Be sure to use the same name when defining the Element Catalog in SMPLJOB2. CA Endevor SCM incorporates the use of an Element Catalog file to support long element names and to improve performance by reducing the volume of I/O operations.

LIBENV

Do not change the value of LIBENV until after running SMPLJOB4, regardless of whether you received an LMP key for this feature.

MACDSN

Names the SOURCE library at your site that contains the CA Endevor SCM macros *(iprfx.iqual.CSIQOPTN*).

PKGDSN

(Required if using packages) Names the package file for your site. The name is supplied during the definition of the package file. For more information, see How to Define and Allocate CA Endevor SCM Libraries (see page 88). Be sure and use this same data set name when allocating the package data set in SMPLJOB1.

PROC

Specifies a value, Y or N, indicating whether your site is licensed for CA Endevor Extended Processors. If your site received an LMP key for Product A9, Extended Processors, enter Y. Otherwise, enter N.

SITEID

Names the 1-character name that identifies your site. This field is used internally to differentiate between sites. The default is 0 (zero).

Important! The SITEID is an integral part of the CA Endevor SCM footprint. Any changes to this parameter for existing CA Endevor SCM applications will result in a footprint-compromised error for each element within that environment.

VIOUNIT

Names the unit name for temporary disk data sets that are stored on a virtual I/O unit.

WRKUNIT

Names the unit name for temporary disk data sets that are *not* stored on a virtual I/O unit.

Note: For more information about all items that appear in the TYPE=MAIN portion of the C1DEFLTS Table, see the *Administration Guide*. Reference that guide when creating your own C1DEFLTS Table.

2-How to Define and Allocate Your Libraries

You must define and allocate the following libraries for the sample application:

- The Package File.
- The Master Control File.
- The Element Catalog.
- The Base, Delta, and Output Libraries.

Use members SMPLJOB1 and SMPLJOB2, which are stored in iprfx.iqual.CSIQSAMP, to allocate these files.

Important! The VSAM files used by CA Endevor SCM, the Master Control files, the Element Catalog, and the package file must be maintained at a two-level index for file integrity purposes. CA Endevor SCM automatically recognizes if a file has less than two index levels and dynamically adjusts it. The next user to access CA Endevor SCM will not encounter a similar problem.

How to Allocate the PACKAGE FILE

The PACKAGE FILE is a VSAM data set that stores all packages built to perform element actions in any of the environments defined in the C1DEFLTS Table. Only one package data set can exist per site.

To allocate the PACKAGE FILE, do the following:

1. Defined the PACKAGE FILE in the C1DEFLTS Table. Be sure the name of the PACKAGE FILE allocated is the same as the name provided in the PKGDSN parameter in the C1DEFLTS Table.

   Note: For more information, see Enter Site-Specific Parameters in the TYPE=MAIN Macro (see page 86).

   This step allocates the following VSAM structure:

Library	Data Set Name
VSAM cluster	*iprfx.iqual.*SMPL.PACKAGE
VSAM cluster containing data	*iprfx.iqual.*SMPL.PACKAGE.DATA
VSAM cluster containing indexes	*iprfx.iqual.*SMPL.PACKAGE.INDEX

2. If you have not already done so, copy your JOBCARD member to the beginning of SMPLJOB1, and then submit the job for execution. Member SMPLJOB1 is stored in *iprfx.iqual*.CSIQSAMP and is used to allocate this file.

   When you run SMPLJOB1, the JCL completes the following steps:

   - Executes IDCAMS to delete any existing package files the first time you run this job.
   - Defines the (new) package file.

Master Control File

An environment in CA Endevor SCM is made up of two stages; a unique MCF is required for each. The MCF values are defined to CA Endevor SCM using the STG1VSAM and STG2VSAM parameters in the TYPE=ENVRNMNT section of the C1DEFLTS Table. The MCF stores the system, subsystem, type, and element definitions for the stage. Be sure the iprfx.iqual components of the Master Control File data set names are the same values as the data set names coded in the C1DEFLTS table.

Element Catalog

CA Endevor SCM incorporates the use of an Element Catalog file to support long element names and to improve performance by reducing the volume of I/O operations. It is identified to CA Endevor SCM by the ELMCATL parameter in the TYPE=MAIN section of the C1DEFLTS Table.

Base, Delta, and Output Libraries

The base, delta, and output libraries are used during CA Endevor SCM execution. Base libraries store the original or current source version of an element. Delta libraries track the differences between the numerous source levels of an element. Source output libraries contain readable copies of an element; for example, JCL, PROC, COPYBOOK, and data parameters. Processor output libraries contain executables; for example, load libraries, object decks, and DBRMs. CA Endevor SCM uses Processor load libraries to store the executable version of a processor.

Note: For more information about base and delta libraries, see the *Administration Guide*. For more information about processors, see the *Extended Processor Guide*.

You can have multiple occurrences of the libraries in the following list, usually one set of libraries for each stage. Only two processor load libraries are allocated-one for each production stage where the processor type is defined.

The base, delta, and output libraries that you can allocate include (but are not limited to) the following:

- Source (base) library
- Delta library
- Copy library
- Object library
- Load library
- Listing library
- Link-edit control card library
- JCL library
- JCL procedure library
- DBRM library (for DB2 users)

How to Run SMPLJOB2

SMPLJOB2 defines the Master Control Files (MCF), the Element Catalog, and any base, delta, and output libraries used by the sample application. It deletes any existing libraries before allocating them.

SMPLJOB2 is found in the CA Endevor SCM JCL library, *iprfx.iqual.*CSIQSAMP. If you have not already done so, copy your JOBCARD member to the beginning of SMPLJOB2, and then submit the job for execution. If the job is interrupted or terminates abnormally, restart the job from the beginning.

When you run SMPLJOB2, the JCL performs the following actions:

1. DELMCF executes IDCAMS to delete any Master Control Files that exist.
2. DEFMCF executes IDCAMS to define the (new) Master Control Files.
3. DELECATL executes IDCAMS to delete the sample Element Catalog, if one exists.
4. DELEIX executes IDCAMS to delete the Element Catalog cross-reference file.
5. DEFECATL executes IDCAMS to define the (new) sample Element Catalog.
6. DEFEIX executes IDCAMS to define the Element Catalog cross-reference file.
7. DELPDS executes IDCAMS to delete the non-VSAM (base, delta, and output) libraries before they are reallocated.
8. DEFPDS executes IEFBR14 to allocate the non-VSAM (base, delta, and output) libraries.

How to Allocate and Initialize ACM Query Files (SMPLACMD)

Important! If your site does not have a license for ACM, skip this step and continue with the Implementation Checklist (see page 92).

You must define and initialize the sample application ACM Root (ACMROOT) and Cross-reference (ACMXREF) files used by the ACM Query Facility (ACMQ) if you specified ASCM=Y. ASCM=Y indicates that you are using the Automated Configuration Manager at your site.

To define and initialize the ACMROOT and ACMXREF files, do the following:

1. Use member SMPLACMD, located in *iprfx.iqual.CSIQSAMP*, to define the Root and Cross-reference data sets.

   The ACMQ Root data set contains the name of each element used as an input component to other elements, as well as all related components. The ACMQ Cross-reference data set contains a record for each component relationship.

These files are defined to CA Endevor SCM in the ACMROOT and ACMXREF parameters in the TYPE=MAIN section of the C1DEFLTS Table. There is only one ACMROOT and ACMXREF pair of data sets per site.

2. Make sure that the name of the allocated ACMROOT and ACMXREF files are the same as the names provided in the ACMROOT and ACMXREF parameters in the C1DEFLTS Table.

   Note: For more information about these parameters, see Enter Site-Specific Parameters in the TYPE=MAIN Macro (see page 86).

Library	Data Set Name
VSAM cluster	*ipfrx*.*iqual*.SMPL.ACMROOT
VSAM cluster	*iprfx.iqual*.SMPL.ACMXREF

Note: For more information about ACMQ, see the *Automated Configuration Option Guide*.

How to Define and Initialize the Root and Cross-reference Data Sets

Use member SMPLACMD, located in *iprfx.iqual.CSIQSAMP*, to define (create) the Root and Cross-reference data sets used by the sample application.

1. Edit member SMPLACMD to add a valid jobcard and verify that the cluster names and the VOLUMES parameter are what you expect. These values were customized according to the values you supplied in installation job BC1JJB03.
2. Follow the instructions in SMPLACMD to further edit the JCL and submit the JOB to create and initialize the sample application root and cross-reference data sets.

Note: After you have finished using the sample application and you are ready to build your production CA Endevor SCM application, use CSIQJCL member BC1JACMD to build your production ACMQ files.

Implementation Checklist

Before implementing the rest of the sample application, ensure that you have installed CA Endevor SCM correctly. Ask yourself the following questions, which reflect typical problems that might occur when installing the sample application.

- Have your LMP keys been installed?
- Have all CA Endevor SCM-supplied libraries been created and populated? Do they contain the correct number of modules? (For example, CSIQAUTH 80+, CSIQLOAD 700+)
- Have the MCFs, Element Catalog/EINDEX, and Package File been created correctly?
- Are TYPE=MAIN, TYPE=ENVRNMNT, and TYPE=END present and in the correct order in the CA Endevor SCM C1DEFLTS Table?

- If the authorized libraries (CSIQAUTH, CSIQAUTU) are not being placed in the linklist, did you add a STEPLIB to the proper JCL skeletons, LOGON procs, and CLISTs?
- If the authorized libraries are in the linklist, did you perform an LLA REFRESH after creating the C1DEFLTS table in Step 1?
- Test system authorization for NDVRC1. Is CSIQAUTH authorized? Depending upon the release of TSO, has the NDVRC1 keyword been added to the proper system tables?
- Can CA Endevor SCM find and load NDVRC1, C1DEFLTS, and any other programs that typically reside in the CSIQAUTH/CSIQLOAD?

3-How to Define the Sample Application Inventory Structure (SMPLJOB3)

You define the sample application inventory structure (systems, subsystems, types) to CA Endevor SCM by executing the Batch Environment Definition Facility (also known as Batch Admin).

Note: For more information about using the Batch Environment Definition Facility, see the *SCL Reference Guide*.

To execute the Batch Environment Definition Facility, do the following:

1. Use the JCL from SMPLJOB3, which is in the library *iprfx.iqual.CSIQSAMP*.
2. If you have not already done so, copy your JOBCARD member to the beginning of SMPLJOB3, and then submit the job for execution.

   In the SMPLJOB3 JCL, notice the last two lines. The second-to-last line refers to member SCLXTND and is uncommented. The last line refers to member SCLBASE and is commented out.
3. If your site is using extended processors, do nothing. Otherwise, uncomment the SCLBASE, ddname ENESCLIN line and comment out the line specifying member SCLXTND by adding //* to the front of that line.

   SCLXTND contains SCL to define system FINANCE, subsystems ACCTPAY and ACCTREC, and several types such as COBOL, ASM, and COPY, as well as several processor groups for each of these types. These definitions are made to both stages of the SMPLTEST and SMPLPROD environments. The FINANCE system is where the source elements used by the sample application are stored. An additional system, ADMIN, is defined to the SMPLPROD environment. This system contains one subsystem and one type - PROCESS. This is where the processors used by the sample application are stored.

   SCL contains SCL similar to SCLXTEND in that it defines the same systems, subsystems, and types. It differs in the respect that no processor groups are defined.

4. If this step does not complete satisfactorily (return code greater than 08), research the error, then rerun SMPLJOB2 before running SMPLJOB3 again.

Note: To quickly find the error message associated with an RC=12, view the job's output in SDSF or another output management utility, page down to the C1MSGS1 output, and on the COMMAND line type *F 'E ' 19 ALL* and press ENTER. An ENBE*nnn*E message is displayed. For more information about this message, see the *Messages and Codes Reference Guide.*

Populate the Sample Application (SMPLJOB4)

You can add processors and source data to the sample application. Use the JCL member SMPLJOB4, which is in the library *iprfx.iqual.CSIQSAMP*. This job adds elements to the application and, if your site is using processors, adds the sample processors and generates the elements.

If you have not already done so, copy your JOBCARD member to the beginning of SMPLJOB4, then submit the job for execution. If this step does not complete successfully, perform a restart at the step where termination occurred.

Note: This job may exceed the output line limit. Submit the job appropriately.

When you run SMPLJOB4, the JCL performs the following actions.

1. Adds processors to the sample application, if your site is using them.
2. Adds elements to the sample application.
3. Moves all elements to SMPLPROD stage 2.

4-How to Verify the Implementation

After you have performed the installation steps and executed the sample application implementation jobs, you need to verify that the sample environments have been created correctly. The following information summarizes the verification process:

1. Be sure that the sample jobs have completed successfully. If a job has failed to execute successfully, you must rerun that job or, in some situations, restart from a step that failed within the job.
2. Start CA Endevor SCM. Check the Environment Selection panel to ensure that the environments SMPLTEST and SMPLPROD appear (see page 95).
3. Execute CONRPT02, the System Inventory Summary and review it to ensure that all sample application elements have been loaded (see page 95).
4. Run foreground (see page 100), batch (see page 102), and ACMQ verification procedures (see page 103).

View the Environment Selection Panel

Ask your system administrator how CA Endevor SCM has been identified to ISPF. Typically, this is done as an ISPF menu option.

Select the CA Endevor SCM option from the appropriate ISPF panel. With the sample application properly implemented, the CA Endevor SCM Environment Selection panel displays the sample environments as options. You should see options for both the Sample Test Environment and the Sample Production Environment.

View the System Inventory Summary Report

To verify that all elements were correctly added to CA Endevor SCM, view CONRPT02, the System Inventory Summary Report, for the SMPLPROD environment. Review the report to verify that the number of elements in SMPLPROD is correct.

To view the System Inventory Summary Report

1. Start CA Endevor SCM using the instructions provided by your site administrator.

   The Primary Options panel appears.

2. Select U (User Menu), and press Enter.

   The User Options Menu appears.

3. Select Option 1, Reports, and press Enter.

   The Reporting Interface panel appears.

4. Select option 1, Master, and press Enter.

   The Master Control File Reports panel appears.

5. Select 02, System Inventory Summary, and enter the following values.

   ENVIRONMENT

   Enter SMPLPROD

   SYSTEM

   Enter an asterisk (*)

   SUBSYSTEM

   Enter an asterisk (*)

   ELEMENT

   Enter an asterisk (*)

TYPE

Enter an asterisk (*)

STAGE

Enter an asterisk (*)

SEARCH ENVIRONMENT MAP

Enter N

6. Press Enter.

   CA Endevor SCM generates JCL to create CONRPT02, the System Inventory Summary, for environment SMPLPROD. The CA Endevor SCM Reporting Interface panel opens, displaying the message, Job Step Created, in the upper right of the panel.

7. Select option S, Submit, and press Enter.

   CA Endevor SCM submits the JCL for execution and displays the message, Job Submitted, in the upper right hand corner of the panel.

   Compare the output from the System Inventory Summary you generate to the following sample report. The values on your report should match the sample report values.

If any discrepancies exist between your report and the sample report, review the output from SMPLJOB4. Look for any error messages that might have occurred. Correct the problems, resubmit SMPLJOB4, and run CONRPT02 again.

Example: View the System Inventory Summary Report

Your System Inventory Summary should appear as follows.Although the numbers in the following report were accurate at the time this guide was published, the numbers in your report may vary slightly.

```
1  Copyright (C) 1986-2010 CA. All Rights Reserved.                                        04/30/10  10:01:04     PAGE     1
 *** PUT YOUR COMPANY NAME HERE ***                  CA Endevor SCM                            RELEASE 14.0  SERIAL B1400C

 C1BR2000: INPUT PARAMETERS:

      REPORT  02 .
      ENVIRONMENT SMPLPROD .
      DAYS        7 .
0C1BR2000: SELECTION SUMMARY:
           REPORT(S)         : 02
0          EXTRACT(S)        :
           ENVIRONMENT       : SMPLPROD
           SYSTEM            : ********
           SUBSYSTEM         : ********
           TYPE              : ********
           STAGE             : *
           ELEMENT           : **********
           DAYS              : 00007
1  Copyright (C) 1986-2010 CA. All Rights Reserved.                                        04/30/10  10:01:04     PAGE     2
 *** PUT YOUR COMPANY NAME HERE ***                  CA Endevor SCM                            RELEASE 14.0  SERIAL B1400C

 C1BR3000: EXTRACT PHASE SUMMARY
0REPORT TITLE                                      RECORDS
 -------------------------------------------------  -------
0CONRPT02:  SYSTEM INVENTORY SUMMARY                     39
0TOTAL NUMBER OF RECORDS FOR REPORTS EXTRACTED           39
0SORT CONTROL PARAMETERS:
0     SORT FIELDS=(5,69,CH,A),SIZE=0000039,EQUALS
      RECORD TYPE=V,LENGTH=(2052,,,0073)
0SORT COMPLETION CODE: 0000
1  Copyright (C) 1986-2010 CA. All Rights Reserved.                                        04/30/10  10:01:04     PAGE     3
 *** PUT YOUR COMPANY NAME HERE ***                  CA Endevor SCM                            RELEASE 14.0  SERIAL B1400C

                                    CONRPT02:  SYSTEM INVENTORY SUMMARY

           ENVIRON     SYSTEM      SUBSYS      TYPE        STAGE  NUMBER OF          TOTAL    AVERAGE # OF    LARGEST # OF
                                                           ID SEQ  ELEMENTS      STATEMENTS      STATEMENTS      STATEMENTS

           SMPLPROD    ADMIN       PROCESS     PROCESS     P    2         5             434              87             149
           --------    --------    --------    --------    ------  --------    ------------
           SMPLPROD    ADMIN       PROCESS     PROCESS                  5             434
           SMPLPROD    ADMIN       PROCESS                              5             434
           SMPLPROD    ADMIN                                            5             434
1  Copyright (C) 1986-2010 CA. All Rights Reserved.                                        04/30/10  10:01:04     PAGE     4
 *** PUT YOUR COMPANY NAME HERE ***                  CA Endevor SCM                            RELEASE 14.0  SERIAL B1400C

                                    CONRPT02:  SYSTEM INVENTORY SUMMARY

           ENVIRON     SYSTEM      SUBSYS      TYPE        STAGE  NUMBER OF          TOTAL    AVERAGE # OF    LARGEST # OF
                                                           ID SEQ  ELEMENTS      STATEMENTS      STATEMENTS      STATEMENTS

           SMPLPROD    FINANCE     ACCTPAY     ASM         P    2         2             103              52              83
           --------    --------    --------    --------    ------  --------    ------------
           SMPLPROD    FINANCE     ACCTPAY     ASM                      2             103
0          SMPLPROD    FINANCE     ACCTPAY     COBOL       P    2         8             390              49             102
           --------    --------    --------    --------    ------  --------    ------------
           SMPLPROD    FINANCE     ACCTPAY     COBOL                    8             390
0          SMPLPROD    FINANCE     ACCTPAY     COPY        P    2         5              35               7              14
           --------    --------    --------    --------    ------  --------    ------------
           SMPLPROD    FINANCE     ACCTPAY     COPY                     5              35
0          SMPLPROD    FINANCE     ACCTPAY     MACRO       P    2         1              55              55              55
           --------    --------    --------    --------    ------  --------    ------------
           SMPLPROD    FINANCE     ACCTPAY                              16             583
0          SMPLPROD    FINANCE     ACCTREC     COBOL       P    2         9             528              59             106
           --------    --------    --------    --------    ------  --------    ------------
           SMPLPROD    FINANCE     ACCTREC     COBOL                    9             528
0          SMPLPROD    FINANCE     ACCTREC     COPY        P    2         9              74               8              15
           --------    --------    --------    --------    ------  --------    ------------
           SMPLPROD    FINANCE     ACCTREC     COPY                     9              74
           SMPLPROD    FINANCE     ACCTREC                             18             602
           SMPLPROD    FINANCE                                         34            1185
           SMPLPROD                                                    39            1619
```

If you ran SMPLJOB3 using SCLBASE as input, your report should look similar to the following report. Although the numbers in the following report were accurate at the time this guide was published, the numbers in your report may vary slightly.

```
1  Copyright (C) 1986-2010 CA. All Rights Reserved.                                        04/30/10  09:44:24     PAGE     1
 *** PUT YOUR COMPANY NAME HERE ***                  CA Endevor SCM                            RELEASE 14.0  SERIAL B1400C

 C1BR2000: INPUT PARAMETERS:

      REPORT  02 .
      ENVIRONMENT SMPLPROD .
      DAYS        7 .
0C1BR2000: SELECTION SUMMARY:
           REPORT(S)         : 02
0          EXTRACT(S)        :
           ENVIRONMENT       : SMPLPROD
           SYSTEM            : ********
           SUBSYSTEM         : ********
           TYPE              : ********
           STAGE             : *
           ELEMENT           : **********
           DAYS              : 00007
1  Copyright (C) 1986-2010 CA. All Rights Reserved.                                        04/30/10  09:44:25     PAGE     2
 *** PUT YOUR COMPANY NAME HERE ***                  CA Endevor SCM                            RELEASE 14.0  SERIAL B1400C

 C1BR3000: EXTRACT PHASE SUMMARY
0REPORT TITLE                                      RECORDS
 -------------------------------------------------  -------
0CONRPT02:  SYSTEM INVENTORY SUMMARY                     34
0TOTAL NUMBER OF RECORDS FOR REPORTS EXTRACTED           34
0SORT CONTROL PARAMETERS:
0     SORT FIELDS=(5,69,CH,A),SIZE=0000034,EQUALS
      RECORD TYPE=V,LENGTH=(2052,,,0073)
0SORT COMPLETION CODE: 0000
1  Copyright (C) 1986-2010 CA. All Rights Reserved.                                        04/30/10  09:44:25     PAGE     3
 *** PUT YOUR COMPANY NAME HERE ***                  CA Endevor SCM                            RELEASE 14.0  SERIAL B1400C

                                      CONRPT02:  SYSTEM INVENTORY SUMMARY

           ENVIRON     SYSTEM      SUBSYS      TYPE        STAGE  NUMBER OF          TOTAL    AVERAGE # OF    LARGEST # OF
                                                           ID SEQ  ELEMENTS      STATEMENTS      STATEMENTS      STATEMENTS

           SMPLPROD    FINANCE     ACCTPAY     ASM         P    2         2             103              52              83
           --------    --------    --------    --------    ------  --------    ------------
           SMPLPROD    FINANCE     ACCTPAY     ASM                       2             103
0          SMPLPROD    FINANCE     ACCTPAY     COBOL       P    2         8             390              49             102
           --------    --------    --------    --------    ------  --------    ------------
           SMPLPROD    FINANCE     ACCTPAY     COBOL                     8             390
0          SMPLPROD    FINANCE     ACCTPAY     COPY        P    2         5              35               7              14
           --------    --------    --------    --------    ------  --------    ------------
           SMPLPROD    FINANCE     ACCTPAY     COPY                      5              35
0          SMPLPROD    FINANCE     ACCTPAY     MACRO       P    2         1              55              55              55
           --------    --------    --------    --------    ------  --------    ------------
           SMPLPROD    FINANCE     ACCTPAY                               16             583
0          SMPLPROD    FINANCE     ACCTREC     COBOL       P    2         9             528              59             106
           --------    --------    --------    --------    ------  --------    ------------
           SMPLPROD    FINANCE     ACCTREC     COBOL                     9             528
0          SMPLPROD    FINANCE     ACCTREC     COPY        P    2         9              74               8              15
           --------    --------    --------    --------    ------  --------    ------------
           SMPLPROD    FINANCE     ACCTREC     COPY                      9              74
           SMPLPROD    FINANCE     ACCTREC                               18             602
           SMPLPROD    FINANCE                                           34            1185
           SMPLPROD                                                      34            1185
```

Verify the Sample Environments (Foreground)

You must verify, in foreground mode, that the sample environments have been loaded correctly.

To verify that the sample environments have been loaded correctly

1. Start CA Endevor SCM using the instructions provided by your site administrator.

   The Primary Options panel appears.

2. Select option 2, Foreground, and press Enter.

   The Foreground Options Menu appears.

3. Select option 4, Generate.

   The Generate Elements panel appears.

4. Enter the following values.

   ENVIRONMENT

   Enter SMPLPROD

   SYSTEM

   Enter FINANCE

   SUBSYSTEM

   Enter ACCTREC

   ELEMENT

   Enter FINARP01

   TYPE

   Enter COBOL

   STAGE

   Enter P

   COMMENT

   Enter VERIFY

   DISPLAY LIST

   Enter Y

   BUILD USING MAP

   Enter Y

   CCID

   Enter SAMPLE

COPYBACK

Enter N

OVERRIDE SIGNOUT

Enter N

5. Press Enter.

   CA Endevor SCM compiles and links program FINARP01. If the sample environments have been installed correctly, the message ELEMENT FINARP01 GENERATED appears in the upper right corner of the Generate Elements panel.

   If the action fails, review the execution report, which is automatically generated, to determine the nature of the problem. If it fails because of errors during the compile or link steps, research the problem using the stored compile and link-edit listings.

View a Stored Listing (Foreground)

You must verify, in foreground mode, that a stored listing has been loaded correctly.

To view a stored listing

1. Start CA Endevor SCM using the instructions provided by your site administrator.

   The Primary Options panel appears.

2. Select option 1, Display, and press Enter.

   The Display Options Menu appears.

3. Select option 2, Footprint, and press Enter.

   The Footprint Display Panel appears.

4. In the OTHER PARTITIONED DATA SET field, enter *iprfx*.*iqual*.SMPLPROD.LISTLIB and press Enter.

   A list of members in that library appears.

5. Tab down to, or issue a locate command for, member FINARP01.

6. Enter L to the left of the member you want to select and press Enter.

   The stored listing is displayed. Review the listing for compile and link errors.

Verify the Sample Environments (Batch)

You must verify, in batch mode, that the sample environments have been loaded correctly.

To verify that the sample environments have been loaded correctly

1. Start CA Endevor SCM using the instructions provided by your site administrator.

   The Primary Options panel appears.

2. Select option 3, Batch, and press Enter.

   The Batch Options Menu appears.

3. Enter the following values:

   PROJECT

   Enter a qualifier for any 80-byte partitioned data set. For example, MCCPE01.

   GROUP

   Enter a qualifier for any 80-byte partitioned data set. For example, PGM.

   TYPE

   Enter a qualifier for any 80-byte partitioned data set. For example, SCL.

   MEMBER

   Enter SCLVER

4. In the Option field, enter 1, and press Enter.

   The SCL Generation menu appears.

5. Select option 4, Generate, and press Enter.

   The Generate Elements panel appears.

6. Enter the following values:

   ENVIRONMENT

   Enter SMPLPROD

   SYSTEM

   Enter FINANCE

   SUBSYSTEM

   Enter ACCTREC

   ELEMENT

   Enter FINARP01

TYPE

Enter COBOL

STAGE

Enter P

COMMENT

Enter VERIFY

CCID

Enter SAMPLE

COPYBACK

Enter N

OVERRIDE SIGNOUT

Enter N

AUTOGEN

Enter N

7. Enter G in the Option field, and press Enter.

   The message SCL GENERATED displays in the upper right corner of the panel.

8. Press PF3 twice.

   The Batch Options Menu appears.

9. In the JOB STATEMENT INFORMATION field, enter your JOBCARD information.

10. Select option 3, Submit, and press Enter.

    The job is submitted for execution and a message displays, confirming that the job was submitted.

    Use SDSF, CA SYSVIEW® Realtime Performance Management, or a similar facility to view the batch job output. Check the last page for a return code of 0000. If the return code is 0000, the job executed successfully. If the return code is not 0000, review the execution report (which is generated automatically) to determine the problem.

5-Perform ACM Query Facility Verification

If you have not already done so, copy your JOBCARD member to the beginning of SMPLACMX in *iprfx.iqual.CSIQSAMP*, and then submit the job for execution. SMPLACMX performs a batch ACM query request that finds all programs that use copybook PAGING.

The output that you see should look similar to the following output:

```
                                E N D E V O R   S Y N T A X   R E Q U E S T   R E P O R T
 REQUESTED BY: USER02

           10:39:04  C1Y0015I  STARTING PARSE OF REQUEST CARDS
            STATEMENT #1
              LIST USING COMPONENTS FOR
              ELEMENT     'PAGING'
              ENVIRONMENT  *
              SYS          *
              SUB          *
              TYPE         COPY
              STAGE NUMBER *
              .
            STATEMENT #2
            EOF STATEMENT GENERATED
           10:39:04  C1Y0016I  REQUEST CARDS SUCCESSFULLY PARSED

                                      E N D E V O R   E X E C U T I O N   R E P O R T
REQUESTED BY: USER02


          10:39:05  C1G0202I  ACTION #1 / STMT #1
          10:39:05  ACMQ203I     LIST USING COMPONENTS FOR
          10:39:05  ACMQ204I        ELEMENT  PAGING     ENVIRONMENT   SMPLPROD
          10:39:05  ACMQ205I        SYSTEM   FINANCE    SUBSYSTEM     ACCTREC
          10:39:05  ACMQ206I        TYPE     COPY       STAGE NUMBER  2
          10:39:05  ACMQ408I     OPTIONS
          10:39:05  ACMQ411I        NO OPTIONS WERE SPECIFIED
          10:39:05  C1G0202I  ACTION #2 / STMT #1
          10:39:05  ACMQ203I     LIST USING COMPONENTS FOR
          10:39:05  ACMQ204I        ELEMENT  PAGING     ENVIRONMENT   SMPLPOD
          10:39:05  ACMQ205I        SYSTEM   FINANCE    SUBSYSTEM     ACCTREC
          10:39:05  ACMQ206I        TYPE     COPY       STAGE NUMBER  1
          10:39:05  ACMQ408I     OPTIONS
          10:39:05  ACMQ411I        NO OPTIONS WERE SPECIFIED
          10:39:05  ACMQ210I  STMT #1     COMPLETED RC = 0000 .

           REQUEST COMPLETED HIGHEST RC = 0

                                   E N D E V O R   A C T I O N   S U M M A R Y   R E P O R T
REQUESTED BY: USER02


                                                                  # ELM  # ELM        STMT ACTION
 ELEMENT      TYPE       ENVIRON    SYSTEM     SUBSYS     STG  USING   USED   RC NUMBER NUMBER
 ----------   --------   --------   --------   --------   ---  ------ ------  -- ------ ------
 PAGING       COPY       SMPLPROD   FINANCE    ACCTREC     2        1      0  00      1      1
 PAGING       COPY       SMPLPROD   FINANCE    ACCTREC     1        4      0  00      1      4
  Copyright (C) 2007 CA. All Rights Reserved.
                                                        ACM QUERY RESULTS
    LVL    ELEMENT      TYPE       ENVIRON    SYSTEM     SUBSYS     STG

     1     PAGING       COPY       SMPLPROD   FINANCE    ACCTREC     2
     2     FINARP01     COBOL      SMPLPROD   FINANCE    ACCTREC     2
 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
     1     PAGING       COPY       SMPLPROD   FINANCE    ACCTREC     1
     2     FINAPP01     COBOL      SMPLPROD   FINANCE    ACCTPAY     2
     2     FINAPP02     COBOL      SMPLPROD   FINANCE    ACCTPAY     2
     2     FINARP02     COBOL      SMPLPROD   FINANCE    ACCTREC     2
     2     FINARP03     COBOL      SMPLPROD   FINANCE    ACCTREC     2
**************************************************** Bottom of Data ***************************
```

Additional Sample Jobs

The CA Endevor SCM library *iprfx.iqual.CSIQSAMP* contains several JCL members that you can use in conjunction with the sample environments. A description of each job follows:

SMPLTRNG

Uses the SCL statements in *iprfx.iqual*.CSIQSAMP (TRNGSCL) to add 20 COBOL programs and copybooks for use in on-site training classes offered by CA. If you want, you can also add these elements for in-house training.

SMPLDEL

Executes IDCAMS and IEFBR14 to delete all the sample application files from disk.

SMPLGENR

Executes CA Endevor SCM Batch SCL to generate all elements in the sample application.

SMPLLIBR

Executes CA Endevor SCM Batch Admin SCL to update type definitions when using CA Librarian.

SMPLPANV

Executes CA Endevor SCM Batch Admin SCL to update type definitions when using CA Panvalet.

SMPLJREL

Executes CA Endevor SCM Batch Admin SCL to define new type definitions that can be used in conjunction with CONRELE in a processor to demonstrate the use of ACMQ to query on user-related information.

SMPLUNLD

Executes the CA Endevor SCM Unload Utility to back up the sample application data.

SMPLRELD

Executes the CA Endevor SCM Reload Utility to restore the sample application data (using Unload utility data as input).

Note: For more information about the Unload/Reload/Validate utility, see the *Utilities Guide*.

Chapter 7: Testing New Releases with Your Data

This section contains the following topics:

How to Test New Releases with Your Data

As an alternative to using the Sample Application, you can use your existing production data for testing upgrades to new releases of CA Endevor SCM. To do this, you must perform several post-installation steps to create a copy of the data on your test system. We recommend that you copy your largest and most active systems. If your site has many environments, you do not have to establish a test version of them all; however, you should use at least two mapped environments.

Using the following steps, you can make copies of your production MCFs and Package File, and make copies of all the base, delta, and output libraries associated with the systems you select. If you are changing the names of these files on the test system, or if you need to establish a test bed on your production system, you must update CA Endevor SCM with the new names.

Perform the following tasks to test new releases with your data:

1. Copy your Element Catalog (see page 108).
2. Copy the Master Control Files (see page 108).
3. Copy the Package File (see page 108).
4. Update your C1DEFLTS Table (see page 109).
5. Synchronize the Element Catalog with your MCFs (see page 109).
6. Copy ACMQ Files (see page 109).
7. Copy Your Production Files (see page 110).
8. Identify Test Libraries (see page 111).
9. Test the New Release (see page 111).

Copy Your Element Catalog

CA Endevor SCM incorporates an Element Catalog file to support long element names and to improve performance by reducing the volume of I/O operations. The Element Catalog file is required and only one is allowed per site. The Element Catalog is identified to CA Endevor SCM through the ELMCATL field in the TYPE=MAIN section of the C1DEFLTS Table.

To create your Element Catalog

1. Run job BC1JRCAT. You can find this member in the *iprfx.iqual.CSIQJCL* that is delivered with the product.
2. Edit the job to ensure that Step2 references your production ELMCATL and EINDEX, and that Step3A, 3B, and 4 references the test catalog you want to create.

Copy the Master Control Files

To test with copies of your MCFs, use member BC1JRMCF located in *iprfx.iqual.CSIQJCL*. This job will REPRO your current MCF files into sequential data sets; create your new MCF, and then REPRO the old data from the sequential files into your new files.

Edit the JCL to ensure that Step2 references the MCFs you are copying from, and the remaining steps reference the test files you want to create.

Copy the Package File

To test using a copy of your existing package file, use member BC1JRPKG from your CA Endevor SCM JCL library. This job backs up your existing data set to a sequential file, deletes and redefines the package file, and then populates the records back into the newly-defined VSAM package file.

To test with a new package file

1. Create a new PDS member using only Step 3A in the JCL.
2. Submit it for execution.
3. Edit the JCL to ensure that Step2 references your production package file, and that Step3A and 3B through 5 references the test package file you want to create.

Update Your C1DEFLTS Table

Before you can use any of your newly copied files, you must first identify them to CA Endevor SCM through the C1DEFLTS table. When your updates are complete, assemble and link-edit the defaults table. Employ an SMP/E USERMOD to assemble and link-edit C1DEFLTS after you customize it. Alternatively, edit the sample JCL BC1JTABL to assemble and link source module C1DEFLTS outside of SMP/E. BC1JTABL is located in the installation library *iprfx.iqual.CSIQJCL*. This stores the defaults table in *iprfx.iqual.CSIQAUTU* as member C1DEFLTS.

The updated table must reside in the authorized library that was established during the installation process. Verify this information with the person who installed the product at your site.

Note: For more information about the C1DEFLTS Table, see How to Edit the Defaults Table (C1DEFLTS) (see page 85).

Synchronize the Element Catalog with Your MCFs

To ensure the Element Catalog that you just copied is properly referenced by the MCFs defined in your C1DEFLTS Table, you need to run job BC1JXCNM. Use member BC1JXCNM in *irpfx.iqual*.CSIQJCL.

First, run the job with the VALIDATE execution parameter. If the job ends with RC=0 in VALIDATE mode, the element catalog and MCFs are synchronized. However, if it returns an RC=4, ensure you are using the correct C1DEFLTS. RC=4 can indicate that a mismatch exists between the Element Catalog and the C1DEFLTS. If your C1DEFLTS is correct, run BC1JXCNM with execution parameter UPDATE to fix the error. In this mode, the program examines all the MCFs defined in the C1DEFLTS table and stamps the MCFs with the Element Catalog data set name.

Copy ACMQ Files

If your site is licensed for Automated Configuration Manager, and you specify ASCM=Y in your C1DEFLTS Table, the ACM query files, ACMROOT and ACMXREF, must be allocated. If you want to make copies of your existing ACMQ files, you can use IDCAMS to REPRO them to a sequential file (attributes FB 4096) and then REPRO them back into your test files. To allocate new files, use job BC1JACMD, and then use BC1JACML to populate them.

Copy Your Production Files

Use one of the following methods to make your production files available for testing:

- Allocate new files (with new names) and copy your current data into them (see page 110).
- If you are testing on an isolated LPAR, you can copy your current files to the test LPAR (see page 110) using their same names.
- Use IBM's DFDSS utility ADRDSSU to COPY (dump) and rename your data files.

Allocate Base, Delta, and Source Processor Output Libraries

You can either allocate new files (with new names) and copy your current data into them, or if you are testing on an isolated lpar you can copy your current files to the test lpar using their same names. Alternatively, you can use IBM's DFDSS utility ADRDSSU to COPY (dump) and rename your data files.

After processing the MCFs, Package, ACMROOT, XREF and Element Catalog files, you need to allocate your data files. Using IEFBR14 or IDCAMS, allocate empty PDS, PDSE, or ELIBS, based on your site's implementation. Copy any base, deltas, source output libraries, processor output, and load libraries associated with the systems and types you plan to use in your testing.

Note: You can use either of these processes – Allocate Base, Delta, and Source Processor Output Libraries or Copy Production Data. Alternatively, you can use IBM's DFDSS utility ADRDSSU to COPY (DUMP) and rename your data files.

Copy Production Data

Using IDCAMS, IEBCOPY, or BC1PNCPY, copy your production files over to the new empty test files.

Note: For more information about these utilities, see the *Utilities Guide*.

When a PDS load module is copied to a PDSE, CA Endevor SCM load module footprint information is retained only if the CA Endevor SCM processor utility BSTCOPY is used. Other utilities, such as IEBCOPY, do not have the capability to copy the *LOADMOD footprint. For more information, see TEC316937, How to Convert AllFusion Endevor Change Manager Load Libraries between PDS and PDSe Formats, on ca.com/support.

Note: You can use either of these processes – Allocate Base, Delta, and Source Processor Output Libraries or Copy Production Data. Alternatively, you can use IBM's DFDSS utility ADRDSSU to COPY (DUMP) and rename your data files.

Identify Test Libraries to CA Endevor SCM

After you have copied all your production files to new empty test files, change the data set names from their production name values to their new test name values. Update the data set names in each stage for each system imported for testing under each environment.

To rename your production file names to your test file names

1. Start CA Endevor SCM using the instructions provided by your site administrator.

   The Primary Options panel appears.

2. Select option 4, Environment, and press Enter.

   The Environment Options menu appears.

3. Select option 8, DATASET, and press Enter.

   The Type Data Set Request panel appears.

4. Enter the name of the system and stage identifier you want to update, enter U in the Option field, and press Enter.

   The Type Data Sets panel appears.

5. To rename a data set, type over the existing name in the DATA SET NAME column.

   Any error messages are displayed under the MSG column. Press F1 to get more information about an error message, or see the *Messages and Codes Reference Guide*.

6. Repeat these steps for both stages of each environment, for each system you copied.

   Note: If you decide not to use your production data files for testing, you have the option of starting from scratch. To do this you need to create all new files, and run Batch Admin (ENBE1000) to build your new environment. For more information about the Batch Admin Utility, see the *SCL Reference Guide*.

Test the New Release

After you have updated all your systems and stages, you are ready to test. Be certain to test your exits, making sure that you reassemble and relink them first.

Appendix A: Installation Worksheets

This section contains the following topics:

Variable Definitions Table

To help you organize and plan for the installation of CA Endevor SCM, use the following table to review and note the detailed definitions for several of the variables that are included in the different installation JCL jobs. The definitions are grouped by type, for example, qualifiers or cluster parameters. Have this table available for reference when you install CA Endevor SCM.

Note: For more information about the definitions for the parameters in the Defaults Table (C1DEFLTS), see the *Administration Guide*.

Variable Definition Type	Variable	Description
Qualifiers	*myiprfx*	Highest-level qualifier used when assigning data set names for installation and execution libraries. This prefix is used for data sets that store CA Endevor SCM software. The software release number should be coded as part of iprfx or iqual. **Note:** The total length of MYIPRFX and MYIQUAL must not exceed 16 characters.
	myiqual	Second-level qualifier used when assigning data sets for installation and execution libraries. This prefix is used for data sets that store CA Endevor SCM software. The software release number should be coded as part of iqual or iprfx. **Note:** The total length of MYIPRFX and MYIQUAL must not exceed 16 characters.

Variable Definition Type	Variable	Description
	myuprfx	Highest-level qualifier used when assigning data set names for CA Endevor SCM user files. This prefix is used for data sets that store user modules. The software release number should not be coded as part of the prefix, because these data sets are independent of the release. **Note:** The total length of MYUPRFX and MYUQUAL must not exceed 16 characters.
	myuqual	Second-level qualifier used when assigning data set names to CA Endevor SCM user files. This prefix is used for data sets that store user modules. The software release number should not be coded as part of the prefix, because these data sets are independent of the release. **Note:** The total length of MYUPRFX and MYUQUAL must not exceed 16 characters.
Sample Application Variables	SYSCLECOMP	The COBOL/LE compiler load library.
	SYSCLERUN	The COBOL/LE runtime library.
	SYSCLELKED	The COBOL/LE Automatic Call (link) library.
	SYSCIICOMP	The COBOL II compiler load library.
	SYSCIILIB	The COBOL II Automatic Call (link) library.

Variable Definition Type	Variable	Description
	MONITOR=COMPONENTS	Indicates whether your site uses the ACM Query Facility. ■ If your site received an LMP key for product AY, Automated Configuration Option (ACM), do *not* change anything. ■ If your site did *not* receive an LMP key, change this value to MONITOR=NONE.
	ADPROC?	Indicates whether your site uses processors. ■ If your site received an LMP key for product A9, Extended Processors, change this value to ADPROC. ■ If your site did *not* receive an LMP key, change this value to ADNOPROC.
	SYSMACLIB	The data set name of the system assembler macro library. For example, SYS1.MACLIB.
	C??	CII or CLE. Indicates the version of COBOL in use at your site. ■ If you are using COBOL/LE, change this value to CLE. ■ If you are using COBOL II, change this value to CII. ■ If your site uses both COBOL/LE and COBOL II, change the value to the version of COBOL you use most often.
	C??NBL	CIINBL or CLENBL. Indicates the default processor group for COBOL elements delivered with the sample application. The C?? portion of this variable should be set to the same value as the variable C??.
Symbolic Device Assignment	mycart	Symbolic device name for cartridge data sets.
	mypdisk	Symbolic device label for permanent disk data sets.

Variable Definition Type	Variable	Description
	mytdisk	Symbolic device label for temporary disk data sets.
Volume Serial Numbers	*mysvolser*	Volume serial number used by SMP/E for DD SMPTLIB.
	mytvolser	Volume serial number of the CA Endevor SCM installation cartridge.
	myvvolser	The volume serial number of the disk used to store VSAM clusters. **Note:** For more information about volumes, see the definition of VOLUMES under Master Control File (MCF) Cluster Parameters.
	mydvolser	The volume serial number of the disk used to store permanent data sets.
Master Control File Cluster Parameters For assistance, contact CA Support at http://ca.com/support.	CYLINDERS (*n n*)	Primary and secondary space allocations, respectively. The default value is (5 1) for Stage 1 and (5 1) for Stage 2, which is sufficient for a system with up to 750 elements in each stage. **Note:** For more information about how to calculate additional space, see Disk Space Requirements Worksheet (see page 119).
Package Data Set Cluster Parameters For assistance, contact CA Support at http://ca.com/support.	CYLINDERS (*n n*)	Primary and secondary space allocations, respectively. The default value is (8 2), which is sufficient for 50 packages averaging 50 elements, with BACKOUT enabled and averaging 4 outputs per element. **Note:** For more information about how to calculate additional space, see Disk Space Requirements Worksheet (see page 119).

Master Control File Allocations for the Sample Application

SMPLJOB4 allocates the following VSAM structures.

Stage	Library	Data Set Name
1	VSAM cluster	*iprfx.iqual.SMPLTEST.MCF*
	Cluster data component	*iprfx.iqual.SMPLTEST.MCF.DATA*
	Cluster index component	*iprfx.iqual.SMPLTEST.MCF.INDEX*
2	VSAM cluster	*iprfx.iqual.SMPLQA.MCF*
	Cluster data component	*iprfx.iqual.SMPLQA.MCF.DATA*
	Cluster index component	*iprfx.iqual.SMPLQA.MCF.INDEX*
3	VSAM cluster	*iprfx.iqual.SMPLEMER.MCF*
	Cluster data component	*iprfx.iqual.SMPLEMER.MCF.DATA*
	Cluster index component	*iprfx.iqual.SMPLEMER.MCF.INDEX*
4	VSAM cluster	*iprfx.iqual.SMPLPROD.MCF*
	Cluster data component	*iprfx.iqual.SMPLPROD.MCF.DATA*
	Cluster index component	*iprfx.iqual.SMPLPROD.MCF.INDEX*

Installation Worksheet

To help you organize and plan the installation of CA Endevor SCM, you can use the following worksheet to record the information and values you will use to install CA Endevor SCM at your site.

Note: If you need clarification about variables when completing this worksheet, see the Variable Definition Table (see page 113).

Variable Definition Type	Variable	Your Site's Value
Qualifiers	*myiprfx*	
	myiqual	
	myuprfx	
	myuqual	

Variable Definition Type	Variable	Your Site's Value
Sample Application Variables	SYSCLECOMP	
	SYSCLERUN	
	SYSCLELKED	
	SYSCIICOMP	
	SYSCIILIB	
	MONITOR=COMPONENTS	
	ADPROC?	
	SYSMACLIB	
	C??	
	C??NBL	
Symbolic Device Assignment	*mycart*	
	mypdisk	
	mytdisk	
Volume Serial Numbers	*mysvolser*	
	mytvolser	
	myvvolser	
	mydvolser	
Master Control File Cluster Parameters	CYLINDERS (*n n*)	
	FREESPACE	**Note:** Do not change the default value of (30 30) for this parameter.
	RECORDSIZE	**Note:** Do not change the default value of (640 1200) for this parameter.
	KEYS	**Note:** Do not change the default value of (28 0) for this parameter.
	SHR	**Note:** Do not change the default value of (3 3) for this parameter.

Variable Definition Type	Variable	Your Site's Value
Package Data Set Cluster Parameters For assistance, contact CA Support at http://ca.com/support.	CYLINDERS (*n n*)	
	FREESPACE	**Note:** Do not change the default value of (30 30) for this parameter.
	RECORDSIZE	**Note:** Do not change the default value of (640 3070) for this parameter.
	KEYS	**Note:** Do not change the default value of (64 8) for this parameter.
	SHR	**Note:** Do not change the default value of (3 3) for this parameter.

Disk Space Requirements Worksheet

The default disk space allocations provided in the installation JCL are large enough to accommodate a pilot project (approximately 500 elements). The information may vary depending on factors such as the average number of lines of code per element, blocking factors, and disk device types. Use the information and examples in the Disk Space Requirements Worksheet to help you determine the requirements for your site. Have this worksheet available for reference when you install CA Endevor SCM. The worksheet is organized as follows:

- Data set name and description
- Default values, where appropriate
- Space for your specification

Before completing the Disk Space Requirements Worksheet, consider the following information:

- It is assumed that your site uses 3390-type disk devices. All SPACE parameters are based on 3390 devices. If your site uses another type of device, you *must* adjust them accordingly.
- All libraries can be defined as operating system partitioned data sets (PDS), or, with the exception of a load library, as CA Panvalet data sets, CA Librarian data sets, or ELIB data sets. If you are defining CA Panvalet or CA Librarian data sets, be sure to allocate storage equivalent to the storage defined in the worksheet.

  Note: For more information about estimating disk space allocation for ELIB, see the *Utilities Guide*.
- When allocating a partitioned data set, be aware that for efficiency, directory blocks are allocated in increments of 45 (which is the number that can fit on a single track of a 3390-type device). In addition, in each library, one directory block exists for every four elements.
- CA Endevor SCM compresses blank spaces from the base, delta, and processor listing libraries (assuming the base and delta libraries are OS/PDS or ELIB files).
- Routine maintenance is required on the libraries described. For more information, see the *Utilities Guide*.
- The number of members stored in a single partitioned data set should not exceed 5,000. Exceeding this number may result in contention problems such as severe disk contention, heavy updating, and excessive wait time for rewriting partitioned data set directories. Up-front planning in this area helps avoid problems. Keep in mind that base and delta libraries can be defined by element types.
- Elements can be stored in reverse delta or forward delta format. The Disk Space Requirements Worksheet provides information for both formats in the descriptions of the base library and the delta library. The CA Endevor SCM sample application uses reverse delta format for storing elements.

Element Catalog Planning

Allocate one 3390 cylinder for every 600 elements in PROD. For the element index, allocate one 3390 cylinder for every 2,000 elements in PROD.

Base Library Planning

Consider the following values when planning a base library:

Data Set Name

*uprfx.uqual.*BASE

Description

This library contains element source as initially defined to CA Endevor SCM. The space requirements are a function of the number of elements (members) to be stored, the number of source lines per element, and the library manager in use. For existing PDS members to be loaded into the base library, use ISPF/PDF option 3.2 (UTILITIES, then DATASETS) to estimate the space required. To calculate the total amount of space needed, multiply the estimated space by an expansion factor (such as 1.2) that is appropriate to the expected library expansion beyond these members.

Format

PDS: DCB=(RECFM=VB,LRECL=259,BLKSIZE=0) (Forward Delta)

PDS: DCB=(RECFM=FB,LRECL=80,BLKSIZE=0) (Reverse Delta)

Space Calculation

To estimate the number of PDS directory blocks needed, assume one directory block exists for every four elements. For efficiency, directory blocks should be allocated in increments of 45 for a 3390-type device (or whatever number can fit on a single track for another type of device). To calculate the number of directory blocks, divide the estimated number of elements (members) to be stored in the library by four. Round that result to an even multiple of 45 (for a 3390-type device). The number should be the same for base and delta libraries.

For example, assume two existing PDS libraries are placed under the control of CA Endevor SCM, and the total space currently occupied by members in the library takes up 100 cylinders. You can allocate a base library of 120 cylinders (100 x 1.2 = 120) to leave 20 percent free space to allow for expansion.

Forward Delta Format

The following values apply if you are storing elements in forward delta format.

	PDS Defaults	Your Site's Specification
Primary (CYL)	80	
Secondary (CYL)	10	
Dir Blk	180	

Reverse Delta Format

The following values apply if you are storing elements in reverse delta format.

	PDS Defaults	Your Site's Specification
Primary (CYL)	90	
Secondary (CYL)	15	

	PDS Defaults	Your Site's Specification
Dir Blk	90	

Delta Library Planning

Consider the following values when planning a delta library:

Data Set Name

*uprfx.uqual.*DELTA

Description

This library contains change levels for elements defined to CA Endevor SCM. The space requirements for the library are a function of the number of elements to be stored, the volatility of the elements (that is, the number of changes that are expected and the extent of those changes), and the library manager in use. A good starting point is a size that is half as large as the base library. The number of directory blocks should be the same as for the base library.

Note: If the Automated Configuration Manager is installed at your site, increase the file size by 33 percent and triple the number of directory blocks. In addition, you cannot use the same library as the Base and Delta library.

Format

PDS: DCB=(RECFM=VB,LRECL=28332,BLKSIZE=0) (Same Format for Forward and Reverse Delta)

Forward Delta Format

The following values apply if you are storing elements in forward delta format.

	PDS Defaults	Your Site's Specification
Primary (CYL)	40	
Secondary (CYL)	10	
Dir Blk	180	

Reverse Delta Format

The following values apply if you are storing elements in reverse delta format.

	PDS Defaults	Your Site's Specification
Primary (CYL)	40	
Secondary (CYL)	15	

	PDS Defaults	Your Site's Specification
Dir Blk	270	

Package Data Set Planning

Consider the following values when planning a package data set:

Data Set Name

*uprfx.uqual.*PACKAGE

Description

This data set stores all packages for the entire site. The size of this file is determined by the number of packages to be stored at any given time and the average size of each. The size can be difficult to calculate because packages vary greatly in size, based on number of actions and whether BACKOUT/BACKIN is enabled. The maximum size of a package record is 3070 bytes.

	Defaults	Your Site's Specification
Primary (CYL)	8	
Secondary (CYL)	2	

Format

VSAM

Processor Load Library Planning (Stage 1)

Consider the following values when planning a processor load library for Stage 1:

Data Set Name

*uprfx.uqual.*PRC1LOAD

Description

This library contains Stage 1 processor load modules. The GPPROCSS processor writes to this library, for elements defined with the element type PROCESS.

The size of this library is a function of the number of processors added to the system with which the library is associated and the number of lines in each processor. Estimate the total number of generate, move, and delete processors to be stored for Stage 1. Allow 1.5 tracks for each processor. For example, for a library targeted to store 15 processors, use a primary allocation of 10 tracks and a secondary allocation of 1 track.

To determine the number of directory blocks needed, assume one directory block exists for every four processors. Then, use the previous information in Base Library to continue.

	PDS Defaults	Your Site's Specification
Primary (CYL)	10	
Secondary (CYL)	5	
Dir Blk	45	

Format

DCB=(RECFM=U,LRECL=0,BLKSIZE=32760)

Listing Libraries Planning (Stage 1)

Consider the following values when planning listing libraries for Stage 1:

Data Set Names

uprfx.uqual.LISTING1

uprfx.uqual.PRC1LIST

Description

These libraries contain listings that correspond to either a Stage 1 user-defined listing library (*uprfx.uqual*.LISTING1) or the Stage 1 processor load modules (*uprfx.uqual*.PRC1LIST).

Size information for each library is discussed in the following section. To determine the number of directory blocks needed for either library, assume one directory block exists for every four listings. Then, use the previous information in Base Library to continue.

Format

PDS: DCB=(RECFM=VBA,LRECL=28332,BLKSIZE=0)

User Listing Library

The CONLIST utility writes to the user listing library, for those elements defined with a type associated with a processor containing this utility. To estimate the size of the user listing library, allocate one track per member. However, the item being stored can greatly influence the overall size of the library. For example, a one-page printout takes up much less space than a 600-page compile.

A good estimate for compile listings is two tracks per member. With a user library targeted to store 100 compile listings, for example, use a primary allocation of 200 tracks and a secondary allocation of 10 tracks.

Processor List Library

The GPPROCSS processor writes to the processor list library, for those elements defined with the type PROCESS. The size of the processor list library is a function of the number of processors added to the system with which the library is associated, and the number of lines in each processor.

Estimate the total number of generate, move, and delete processors to be stored for Stage 1. Allow one track for each processor. For example, for a library targeted to store 15 processors, use a primary allocation of 15 tracks and a secondary allocation of five tracks.

The following table contains the defaults.

	PDS Defaults (PRC1LIST)	PDS Defaults (LISTING1)	Your Site's Specification (PRC1LIST)	Your Site's Specification (LISTING1)
Primary (CYL)	10	200		
Secondary (CYL)	5	10		
Dir Blk	45	45		

Data Set (MCF) Planning (Stage 1)

Consider the following values when planning a data set for Stage 1:

Data Set Name

uprfx.uqual.STAGE1

Description

This data set contains the Stage 1 Master Control File entries. The size of this file is based on the number of systems, subsystems, elements, types, and processor groups to be stored at Stage 1.

	PDS Defaults	Your Site's Specification
Primary (CYL)	5	
Secondary (CYL)	1	

Format

VSAM

COPYLIB Planning (Stage 1)

Consider the following values when planning a COPYLIB for Stage 1:

Data Set Name

uprfx.uqual.COPYLIB1

Description

This library contains the latest full source of each element created during output management at Stage 1. The size of this library is a function of the exact amount of space the members consume outside of CA Endevor SCM.

For existing partitioned data set members, use the ISPF/PDF option 3.2 (UTILITIES, then DATASETS) to estimate the space required. Multiply the space required by an expansion factor (such as 1.2) that is appropriate to the expected library growth beyond these members.

To determine the number of directory blocks needed, assume one directory block exists for every four members. Then, use the previous information in Base Library to continue.

Note: If you are using forward deltas, a source output library must be defined. In addition, if you are using reverse deltas, defining a source output library is optional. The base image library performs the same function as the source output library.

	Your Site's Specification
Primary (CYL)	
Secondary (CYL)	

Format

DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)

Processor Load Library Planning (Stage 2)

Consider the following values when planning a processor load library for Stage 2:

Data Set Name

uprfx.uqual.PRC2LOAD

Description

This library contains Stage 2 processor load modules. Determine the size and number of directory blocks for this file as you would for the Stage 1 library, *uprfx.uqual*.PRC1LOAD. For more information, see Processor Load Library Planning (Stage 1).

	PDS Defaults	Your Site's Specification
Primary (CYL)	10	
Secondary (CYL)	5	
Dir Blk	45	

Format

DCB=(RECFM=U,LRECL=0,BLKSIZE=32760)

Listing Libraries Planning (Stage 2)

Consider the following values when planning listing libraries for Stage 2:

Data Set Names

uprfx.uqual.LISTING2

uprfx.uqual.PRC2LIST

Description

These libraries contain listings that correspond to a Stage 2 user-defined listing library *(uprfx.uqual*.LISTING2) or the Stage 2 processor load modules (*uprfx.uqual*.PRC2LIST).

Determine the size and number of directory blocks for this file as you would for the Stage 1 libraries, *uprfx.uqual*.LISTING1 and *uprfx.uqual*.PRC1LIST. For more information, see Listing Libraries Planning (Stage 1).

	PDS Defaults (PRC2LIST)	PDS Defaults (LISTING2)	Your Site's Specification (PRC2LIST)	Your Site's Specification (LISTING2)
Primary (CYL)	10	200		
Secondary (CYL)	5	10		

	PDS Defaults (PRC2LIST)	PDS Defaults (LISTING2)	Your Site's Specification (PRC2LIST)	Your Site's Specification (LISTING2)
Dir Blk	45	45		

Format

PDS: DCB=(RECFM=VBA,LRECL=28332,BLKSIZE=0)

Data Set (MCF) Planning (Stage 2)

Consider the following values when planning a data set for Stage 2:

Data Set Name

uprfx.uqual.STAGE2

Description

This data set contains the Stage 2 Master Control File entries. Determine the size of this file as you would for the Stage 1 data set, *uprfx.uqual*.STAGE1. For more information, see Data Set (MCF) Planning (Stage 1).

The following table contains the defaults:

	Defaults	Your Site's Specification
Primary (CYL)	5	
Secondary (CYL)	1	

Format

VSAM

COPYLIB Planning (Stage 2)

Consider the following values when planning a COPYLIB for Stage 2:

Data Set Name

uprfx.uqual.COPYLIB2

Description

This library contains the latest full source of each element created during output management at Stage 2. Determine the size of this library as you would for the Stage 1 library, *uprfx.uqual*.COPYLIB1. For more information, see COPYLIB Planning (Stage 1).

Note: If you are using forward deltas, a source output library must be defined. In addition, if you are using reverse deltas that are *not* encrypted, defining a source output library is optional. The base image library performs the same function as the source output library.

	Your Site's Specification
Primary (CYL)	
Secondary (CYL)	

Format

DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)

Appendix B: Using Rapid Implementation

This section contains the following topics:

How to Set Up a Rapid Implementation Configuration

Important: Instead of using the Rapid Implementation process, we recommend that you use the Best Practice Implementation. For more information about the Best Practice Implementation, see the *Scenario Guide*.

If this is the first time CA Endevor SCM is being installed at you site, you can use the Rapid Implementation steps with minimum customization to set up a fully scalable software development environment that is complete and includes a lifecycle and inventory structure. To set up a Rapid Implementation configuration, use one of the following methods:

- If you used Mainframe Software Manager to install the product files, you can use the Software Configuration Service to automate the process of setting up a Rapid Implementation. Software Configuration Services automates the configuration of CA products after they have been installed using Mainframe Software Manager. The Software Configuration Services automated configuration for CA Endevor SCM leads you through all the tasks necessary to implement the Rapid Implementation.
- Use the supplied files to manually set up the Rapid Implementation configuration. Complete the tasks described in the appendix "Using the Rapid Implementation," beginning with How to Perform the Pre-Implementation Tasks (see page 135).

Rapid Implementation

After CA Endevor SCM is installed, your site's systems programmer or CA Endevor SCM administrator can use Rapid Implementation to perform a basic implementation of a predefined environment with minimum customization. The Rapid Implementation process can also be used by CA services personnel or a CA partner.

Rapid Implementation is a series of steps that allow you to perform a standard implementation of CA Endevor SCM using a predefined environment. CA Endevor SCM is a powerful tool with great flexibility that encompasses many processes, designs, implementations, and abilities. After CA Endevor SCM is installed, you can use Rapid Implementation to set up an implementation, with minimum customization, to get you started using CA Endevor SCM. The tasks outlined in this appendix define a simple, but fully scalable software development environment that is complete and includes a lifecycle and inventory structure.

Regardless of the size or complexity of your operation, the Rapid Implementation process gets you started using CA Endevor SCM quickly and easily. After you gain experience using CA Endevor SCM, you can modify, scale, or customize the basic implementation. You can consider changing parameters, turning on optional features, or adding additional environment definitions to the configuration. Rapid Implementation is completely flexible; modifications and additions to the standard implementation can be made at any time after you complete the steps in this guide, or at the time you perform Rapid Implementation. However, if you make extensive customization changes during the Rapid Implementation, the time it takes to complete your implementation will increase.

How Rapid Implementation Works

The Rapid Implementation includes a standard set of jobs that make use of pre-existing CA Endevor SCM utilities that you can use to rapidly create and test an environment in CA Endevor SCM.

Note: Prior to performing the Rapid Implementation, CA Endevor SCM must be installed.

1. One setup job sets your site-specific parameters in the supplied members.
2. After the setup job is complete, the jobs are ready to be executed to configure the CA Endevor SCM environment.
3. You load test data into the environment.
4. Then using the test data, you verify that the configured environment is properly set up.

The Standard Environment

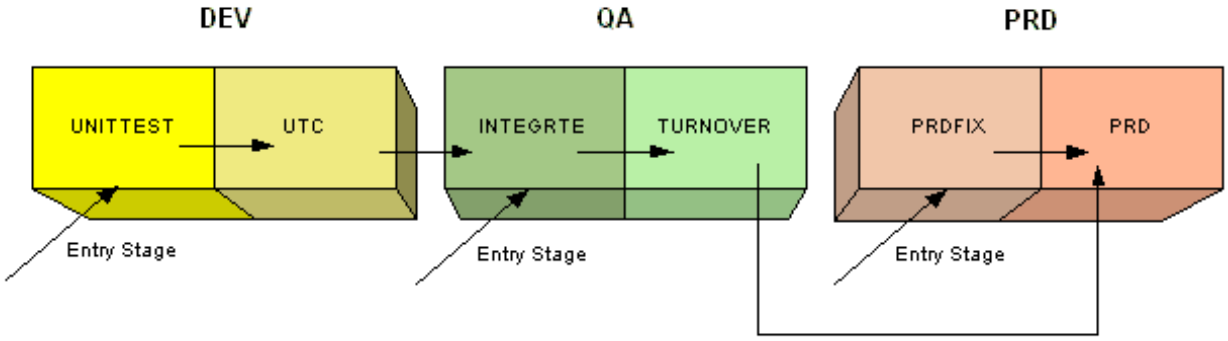
Rapid Implementation adds a predefined lifecycle and a predefined inventory structure that includes systems, subsystems, types, processors, and processor groups.

Predefined Lifecycle

The predefined life cycle includes the following environments, each having two stages:

- DEV—the development environment
  - Stage 1 ID = 1, Name = UNITTEST
  - Stage 2 ID = 2, Name = UTC (Unit Test Complete)
- QA—the quality assurance environment
  - Stage 1 ID = 1, Name = INTEGRTE
  - Stage 2 ID = 2, Name = TURNOVER
- PRD—the production environment
  - Stage 1 ID = 1, Name = PRDFIX
  - Stage 2 ID = 2, Name = PRD

The following diagram illustrates the life cycle implemented by the Rapid Implementation. The arrows between the stages indicate how the stages are mapped to move your software inventory from stage to stage. The mapping is predefined in the setup files.



Predefined Inventory Structure

Rapid Implementation adds a predefined inventory structure, which includes the following systems and subsystems.

Note: For a list of the types, processor groups, and processors provided in the predefined environment, see Supplied Types, Processor Groups, and Processors (see page 173).

Systems

All environments contain the following systems:

COMM@SYS

A system for which the client site chooses the name prior to running the Rapid Implementation setup job. This system is intended to contain components that are common across all systems.

APPL@SYS

A system for which the client site chooses the name prior to running the Rapid Implementation setup job. This usually refers to a product or application name.

The PRD (production) environment also contains the following system:

EA

A system used for CA Endevor SCM administration and includes several members of the JCL data set. We recommend that the system name of EA not be changed.

Note: Optional jobs are provided to allow you to create additional systems.

Subsystems

The following subsystems are included:

PROCESS

Subsystem defined for the EA system for processors.

TABLES

Subsystem defined for the EA system for tables.

COMM@SBS

A subsystem for which the client site chooses the name prior to running the Rapid Implementation setup job. This subsystem is defined for the COMM@SYS system.

APPL@SBS

A subsystem for which the client site chooses the name prior to running the Rapid Implementation setup job. This subsystem is defined for the APPL@SYS system.

Note: Optional jobs are provided to allow you to create additional subsystems.

Rapid Implementation Source Files

Two files are delivered as part of the CA Endevor SCM SMP/E installation process.

- **The Rapid Implementation JCL library**, *iprfx.iqual.CSIQJCLR*, contains jobs, CA Endevor SCM table source, VSAM definition parameters, and SCL syntax members required for the Rapid Implementation. With the exception of the @@UPDJCL member, none of the members in this data set get modified. A copy of this data set is created in the @@UPDJCL job. The members in the copy do get modified to site-specific values. The name of the copy is *uprfx.*RI.JCLLIB.
- **The Rapid Implementation Processor library**, *iprfx.iqual.CSIQPRCR*, contains the source for all processors installed by Rapid Implementation. The members in this data set do not get modified. A copy of this data set is created in the @@UPDJCL job. The members in the copy do get modified to site-specific values. The name of the copy is *uprfx.*RI.PROCSSRS.

The purpose of the copies is to let you start over if you need to do that.

Note: For a list of all the members delivered in the Rapid Implementation JCL library, see Rapid Implementation JCL Library Contents.

How to Perform the Pre-Implementation Tasks

Before you can begin running the Rapid Implementation jobs, you must list the site-specific parameters you will use during the implementation tasks. The pre-implementation tasks also include using CA Endevor SCM security features to secure your software inventory. The following pre-implementation tasks are required:

1. Complete the @@UPDJCL Parameters Worksheet.
2. Verify that the CA Endevor SCM libraries are authorized.
3. Secure data sets using the Alternate ID.
4. Set up access security rules using the security interface (ESI).

1-Complete the Worksheet for Site-Specific Parameters

Enter the values your site plans to use in the @@UPDJCL Parameters Worksheet. You will use these values when you edit the @@UPDJCL member. The @@UPDJCL job updates all members of the Rapid Implementation JCL and processor libraries with these parameters.

@@UPDJCL Parameters Worksheet

Use the following worksheet to list the parameters your site plans to use during the Rapid Implementation.

Your Site's Value	Parameter	Description
	APPL@SBS	Name of subsystem to be created within the APPL@SYS system.
	APPL@SB2	(Optional) Name of a second subsystem to be created within the APPL@SY2 system.
	APPL@SYS	Name of system to be created.
	APPL@SY2	(Optional) Name of a second system to be created.
	COB@TYP	Default COBOL type in use: COBII for COBOL II or COBLE for COBOL LE.
	COMM@SBS	Name of common subsystem to be created within the COMM@SYS system.
	COMM@SYS	Name of the common system to be created.
	CU@NAME	Your company name (50 character maximum).
	DSN@TYPE	Type of file used for base, delta, and all other CA Endevor SCM files. The options are LIBRARY (for PDSE) or PDS (for PDS). The default is LIBRARY. We recommend PDSE data set file type.
	IPRFX	This value must match the value specified for the base CA Endevor SCM installation or runtime libraries.
	IQUAL	This value must match the value specified for the CA Endevor SCM installation or runtime libraries.
	JOB@ACCT#	Job accounting information required on the job cards.
	JOB@NAME	Job name associated with the jobs.
	RACF@UID	Alternate ID as selected by the client. **Note:** For more information on the Alternate ID, see the *Security Guide*.

Your Site's Value	Parameter	Description
	SAND@BOX	(Optional) Name of a sandbox subsystem to be created in the DEV environment for the APPL@SYS system. You can use any valid subsystem name. If a sandbox is to be used by a particular developer, then we recommend you use the developer's user ID as the name. If multiple developers are sharing a sandbox, then name the sandbox according to the project, feature, or CCID that the group is working on. NEXT SUBSYSTEM is equal to APPL@SBS.
	SYSCLECOMP	The COBOL/LE compiler load library. **Note:** For more information, see the Installation Guide.
	SYSCLELKED	The COBOL/LE Automatic Call (link) library. **Note:** For more information, see the Installation Guide.
	SYSCLERUN	The COBOL/LE runtime library. **Note:** For more information, see the Installation Guide.
	SYSCIICOMP	The COBOL II compiler load library. **Note:** For more information, see the Installation Guide.
	SYSCIILIB	The COBOL II Automatic Call (link) library. **Note:** For more information, see the Installation Guide.
	UPRFX	This prefix is used as the high-level data set qualifier for data sets that are created by the CA Endevor SCM Rapid Implementation jobs.

2-Verify the CA Endevor SCM Table Data Set

The *iprfx.iqual.*CSIQAUTU is used to store the CA Endevor SCM tables (C1DEFLTS, ESYMBOLS, and so on). This data set must be concatenated in front of the CA Endevor SCM *iprfx.iqual.*CSIQAUTH used to store CA Endevor SCM authorized programs. Verify that both of these libraries are set up and were APF authorized during the CA Endevor SCM installation process.

3-Secure Your Data Sets

Important! We recommend that you use the Alternate ID to secure your data sets, however you can use CA ACF2 or CA Top Secret program pathing. For more information, see the chapter "Implementing Data Set Security" in the *Security Guide*.

The CA Endevor SCM Alternate ID allows you to secure CA Endevor SCM data sets from updates outside of CA Endevor SCM while allowing users to perform functions that update those data sets from within CA Endevor SCM. Prior to any action that accesses a CA Endevor SCM data set, the Alternate ID is swapped for the user's ID and the access is performed using the Alternate ID.

Typically, you should secure a high–level qualifier as READ/WRITE for CA Endevor SCM, READ/WRITE/UPDATE for the Administrator, and READ for everyone else.

CA Endevor SCM data sets include (but are not limited to) the following:

- Base Libraries
- Delta Libraries
- Package Data Set
- MCF Data Sets
- Element Catalog
- Element Catalog EINDEX
- ACM Root and XREF Libraries
- Source Output
- Processor Output

Note: Ask the systems programmer, who installed CA Endevor SCM at your site, for the data set names.

To secure your data sets

1. Define a user ID for CA Endevor SCM to use as an Alternate ID to your security software product (CA Top Secret, CA ACF2 or RACF).
2. Grant the Alternate ID CONTROL access to all CA Endevor SCM VSAM files and UPDATE access to Source Output and any other libraries and files that are to be written to. A single HLQ (uprfx) is used in the Rapid Implementation setup job, so you can grant access to all the data sets using one rule.
3. Supply the Alternate ID to the RACFUID= parameter in your C1DEFLTS table.

Example: Grant Alternate ID Access to Data Sets

If all your CA Endevor SCM data sets are under **one** high–level qualifier, you can grant the Alternate ID access to all the data sets using **one** rule. The CA ACF2 rule is shown next:

```
uprfx.** R(A) W(A) C(A)
```

Note: For more information on the Alternate ID, see the *Security Guide*.

4-Secure Access to Functions

The External Security Interface (ESI) allows you to establish security rules for access to your Environments, Systems, Subsystems, Actions, and so on, within CA Endevor SCM. This is accomplished by defining pseudo data set names and functional access levels in the ESI table (BC1TNEQU) that are tested against the user's access level through the SAF (Security Authorization Facility) interface.

Example: Use the External Security Interface for Functional Access

This example shows how to give developers access to browse (display) elements, listings, packages, and so on, in any environment; perform various update actions (add, delete, generate, and so on) in the DEV environment only, but no access to perform environment changes or archive actions in any environment.

Assume that you have three environments, DEV, QA and PRD and you want the developers to have complete access to the DEV environment and read access to the QA & PRD environments.

This sample BC1TNEQU table includes definitions for three different levels of authority required for three groups of actions in the function equates portion of the table:

```
*********************************************************
ACCESS FOR DEVELOPERS
*********************************************************
 FUNCEQU SAFAUTH=READ,
       C1ACTNS=(DISPLAY,PDISPLAY)
*********************************************************
 ACCESS FOR DEVELOPERS
*********************************************************
 FUNCEQU SAFAUTH=UPDATE,
       C1ACTNS=(ADD,DELETE,
       GENERATE,MOVE,
       PBACKOUT,PCAST,PCOMMIT,PCREATE,PDYNAMIC,
       PEXECUTE,PLIST,PMODIFY,PREVIEW,PSHIP,PUTILITY,
       RETRIEVE,SIGNIN,SIGNOVR,UPDATE)
*********************************************************
 ACCESS FOR ENDEVOR ADMIN
*********************************************************
 FUNCEQU SAFAUTH=CONTROL,
       C1ACTNS=(ENVRNMGR,ARCHIVE)
```

In the name equates portion of the table, which forms the pseudo data set name to be tested, the action initiation parameters are set as follows:

```
NAMEQU ACTION_INITIATION,
      L1=('C1'),
      L2=(ENVIRONMENT),
      L3=(SYSTEM),
      L4=(SUBSYSTEM),
      L5=(MENUAUTH)
```

Note: The quoted values in the NAMEQU entries are fixed values; the unquoted values are variables that are resolved at the time an action is attempted.

This sample BC1TNEQU table shows that if developers are given Read access to C1.* and update access to C1.DEV.*, they will be able to browse (display) elements, listings, packages, and so on, in any environment; perform various update actions (add, delete, generate, and so on) in the DEV environment only and will not be able to perform environment changes or archive actions in any environment.

The L1 qualifier in this example should not be an actual high–level qualifier that is in use in your shop. Although you may want developers to be able to update source in the DEV environment within CA Endevor SCM, you do not want them to be able to do so outside of CA Endevor SCM. If the actual physical files for DEV are BST.DEV.*, you should give them only read access to this prefix, so that they will only be able to read these files when accessing them outside of CA Endevor SCM.

Note: For more information, see the chapter "Enabling External Security Interface" in the *Security Guide*.

How to Set Up the Standard Environment

Rapid Implementation provides a series of jobs that allow you to perform a standard implementation of CA Endevor SCM using a predefined environment.

To set up the standard environment, perform the following procedures:

1. Update the Rapid Implementation members for your site.
2. Define the EA system files.
3. Process the CA Endevor SCM tables.
4. Define the ACMQ files.
5. Define the package file.
6. Define the Element Catalog and index files.
7. Define the Master Control files.
8. Define the DEV, QA, and PRD APPL@SYS files.
9. Define the DEV, QA, and PRD COMM@SYS files.
10. Define the EA system.
11. Add processors to the EA system.
12. Define the EA system types.
13. Add the CA Endevor SCM tables into the EA system.
14. Promote the CA Endevor SCM tables and processors.
15. Copy the Table Load Modules to the CSIQAUTU Library.
16. Define the APPL@SYS system in the PRD environment.
17. Define the APPL@SYS system in the QA environment.
18. Define the APPL@SYS system in the DEV environment.
19. Define the COMM@SYS system in the PRD environment.

20. Define the COMM@SYS system in the QA environment.
21. Define the COMM@SYS system in the DEV environment.

Note: If you need to restart the Rapid Implementation process, optional jobs are provided to allow you to delete the Rapid Implementation data sets already allocated. For more information, see Optional Jobs to Delete the Rapid Implementation Files.

1-Update the Rapid Implementation Members for Your Site

To customize the implementation for your site, the Rapid Implementation members must be edited with your site-specific parameters.

To update the Rapid Implementation members for your site

1. Edit the @@UPDJCL member located in the iprfx.iqual.CSIQJCLR library and change the values containing underscores with the values you recorded in the @@UPDJCL Parameters Worksheet. This also applies to the four parameters in the SET statement.

   Your site's data set name prefixes, CA Endevor SCM parameters, and definitions are added to the @@UPDJCL member.

   Note: For more information, see @@UPDJCL Parameters Worksheet.

2. Verify that member @@UPDJCL contains a valid job card and submit the job.

   The @@UPDJCL job makes copies of the two libraries iprfx.iqual.CSIQJCLR and *iprfx.iqual.CSIQPRCR* and names the copies *uprfx*.RI.JCLLIB and *uprfx*.RI.PROCSSRS.

   The job also updates all members of *uprfx*.RI.JCLLIB and *uprfx*.RI.PROCSSRS with the parameters you specified in the @@UPDJCL member. Changes made by the @@UPDJCL job are made to the copies, not to the original data sets iprfx.iqual.CSIQJCLR and iprfx.iqual.CSIQPRCR. This allows you to re–execute the @@UPDJCL if you need to start over again.

Note: The allocation PROC included with Rapid Implementation, SCMMALLO, supports the use of PDS or PDSE data sets. The global change job, @@UPDJCL sets the DSN@TYPE parameter value (PDS or LIBRARY) chosen by the user. Regardless of which value is chosen, the SPACE parameter on the DD statement allocations contain a directory block sub–parameter. This sub–parameter is ignored by z/OS when the FILETYPE=LIBRARY (PDSE data set) is specified.

Review the VSAM Definition Parameter Members

Some of the jobs in the JCL file *uprfx*.RI.JCLLIB define VSAM files. Each of these jobs points to a member in the *uprfx*.RI.JCLLIB file that contains VSAM definition parameters (member names starting with VSD). Verify that the VSAM definition parameter members satisfy the VSAM definition requirements at your site. For example, your site might require the STORCLAS parameter to direct the data set to a certain set of disk packs.

2-Define the EA System Files

Job @JOB0010 defines all the data sets required to set up the EA Administrative System.

To define the EA system files

1. Verify that the @JOB0010 job card information is correct.
2. Submit job @JOB0010 for execution.

   All the data sets required to set up the EA Administrative System are defined. IEFBR14 is used to define these data sets. The data set attributes vary, depending on the purpose of each data set.

3-Process the CA Endevor SCM Tables

Job @JOB0020 installs the primary CA Endevor SCM tables required for any CA Endevor SCM job to be executed and copies the global type sequence table to the PARMLIB.

To process the CA Endevor SCM tables

1. Verify that the @JOB0020 job card information is correct.
2. Submit job @JOB0020 for execution.

   Job @@JOB0020 does the following:

   - Assembles and link edits the C1DEFLTS and the ESYMBOLS tables into the user authorized library, *iprfx.iqual*.CSIQAUTU, that was defined as part of the CA Endevor SCM installation. Existing copies of these tables are replaced.
   - Adds the global type sequence table, GLBLTYPE, into the *uprfx*.P2.PARMLIB that was defined by @JOB0010.

     Note: A global type sequence table, member name GLBLTYPE, is delivered in the *uprfx*.RI.JCLLIB library.

4-Define the ACMQ Files

Job @JOB0030 defines the ACM root and cross-reference data sets required for the ACM Query facility.

To define the ACMQ files

1. Verify that the @JOB0030 job card information is correct.
2. Verify the @JOB0030 VSAM definition parameter members. The VSAM definition parameters required by this job are located in the *uprfx*.RI.JCLLIB library, member names VSDACMR and VSDACMX.
3. Submit job @JOB0030 for execution.

   This defines the root and cross-reference data sets required for the ACM Query facility.

Note: For more information, see the *Automated Configuration Option Guide*.

5-Define the Package File

Job @JOB0040 defines the package data set required for the package facility.

To define the package file

1. Verify that the @JOB0040 job card information is correct.
2. Verify the @JOB0040 VSAM definition parameter members. The VSAM definition parameters required by this job are located in the *uprfx*.RI.JCLLIB library, member name VSDPKG.
3. Submit job @JOB0040 for execution.

   Job @@JOB0040 defines the package data set required for the package facility.

6-Define the Element Catalog and Index Files

Job @JOB0050 defines the element catalog and element index data sets.

To define element catalog and index files

1. Verify that the @JOB0050 job card information is correct.
2. Verify the @JOB0050 VSAM definition parameter members. The VSAM definition parameters required by this job are located in the *uprfx*.RI.JCLLIB library, member names VSDELMC and VSDEINDX.
3. Submit job @JOB0050 for execution.

   This defines the element catalog and index files.

7-Define the Master Control Files

Job @JOB0060 defines all the Master Control Files (MCFs) data sets. One MCF is defined for each of the stages in the lifecycle.

To define the master control files

1. Verify that the @JOB0060 job card information is correct.
2. Verify the @JOB0060 VSAM definition parameter members. The VSAM definition parameters required by this job are located in the *uprfx*.RI.JCLLIB library, member names VSMCFD1, VSMCFD2, VSMCFQ1, VSMCFQ2, VSMCFP1 and VSMCFP2.
3. Submit job @JOB0060 for execution.

   The Master Control Files are defined. One file is defined for each of the stages in the predefined lifecycle.

8-Define the DEV, QA, and PRD APPL@SYS Files

Job @JOB0070 defines the base and the delta data sets for the DEV, QA and PRD environments for the APPL@SYS system. One base library is defined for each of the types and one delta library is defined to for each system and environment combination.

To define the DEV, QA, and PRD APPL@SYS files

1. Verify that the @JOB0070 job card information is correct.
2. Submit job @JOB0070 for execution.

   This defines the DEV, QA, and PRD APPL@SYS files. IEFBR14 is used to define these data sets. The data set attributes vary, depending on the type of data the data set will contain.

9-Define the DEV, QA, and PRD COMM@SYS Files

Job @JOB0080 defines the base and the delta data sets for the DEV, QA and PRD environments for the COMM@SYS system.

To define the DEV, QA, and PRD COMM@SYS files

1. Verify that the @JOB0080 job card information is correct.
2. Submit job @JOB0080 for execution.

   The DEV, QA, and PRD COMM@SYS files are defined. One base library is defined for each of the types and one delta library is defined to support each system and environment combination. IEFBR14 is used to define these data sets. The data set attributes vary, depending on the type of data the data set will contain.

10-Define the EA System

Job @JOB0100 defines the EA system. The CA Endevor SCM software is executed to perform this step.

To define the EA system

1. Verify that the @JOB0100 job card information is correct.
2. Submit job @JOB0100 for execution.

   The SCL syntax required to define the system, subsystems, types and processor groups used by this job are located in the *uprfx*.RI.JCLLIB library, member names SCLDEF1. This job defines the EA system.
3. View the C1MSGS1 and C1MSGS2 reports produced by this job to verify the actions were successfully executed.

Example: C1MSGS2 Summary Report for @JOB0100

```
Action                    Object          RC Location
------------------------- --------------- -- ----------
DEFINE SYSTEM             EA              0  To Env PRD
DEFINE SUBSYSTEM          PROCESS         0  To Env PRD
DEFINE SUBSYSTEM          TABLES          0  To Env PRD
DEFINE TYPE               PROCESS         0  To Env PRD
DEFINE PROCESSOR GROUP    PROCESS         0  To Env PRD
DEFINE TYPE               PROCESS         0  To Env PRD
DEFINE PROCESSOR GROUP    PROCESS         0  To Env PRD
```

11-Add Processors to the EA System

Job @JOB0110 adds the processors into the EA system that was defined by @JOB0100. The CA Endevor SCM software is executed to perform this step.

To add processors to the EA system

1. Verify that the @JOB0110 job card information is correct.
2. Submit job @JOB0110 for execution.

   The SCL syntax required by this job is located in the *uprfx*.RI.JCLLIB library, member name SCLACT1. The processor source required by this job is contained in the data set *uprfx*.RI.PROCSSRS.

   This job adds processors to the EA system.
3. View the C1MSGS1 and C1MSGS2 reports produced by this job to verify that the actions were successfully executed.

Example: C1MSGS2 Summary Report for @JOB0110

```
                    PROC  NDVR
ACTION    ELEMENT    RC    RC    +--- FROM INFORMATION ---+
ADD       DELACMOP   0000  0000  DSNAME=uprfx.RI.PROCSSRS
ADD       DPDSMBR    0000  0000  DSNAME=uprfx.RI.PROCSSRS
ADD       DSANDBOX   0000  0000  DSNAME=uprfx.RI.PROCSSRS
ADD       GASM       0000  0000  DSNAME=uprfx.RI.PROCSSRS
ADD       GASMMAP    0000  0000  DSNAME=uprfx.RI.PROCSSRS
ADD       GCOB       0000  0000  DSNAME=uprfx.RI.PROCSSRS
ADD       GLINK      0000  0000  DSNAME=uprfx.RI.PROCSSRS
ADD       GPDSMBR    0000  0000  DSNAME=uprfx.RI.PROCSSRS
ADD       GSANDBOX   0000  0000  DSNAME=uprfx.RI.PROCSSRS
ADD       MPDSMBR    0000  0000  DSNAME=uprfx.RI.PROCSSRS
```

12-Define the EA System Types

Job @JOB0120 defines the types, processor groups, and processor symbolics for the EA system in the PRD environment. The CA Endevor SCM software is executed to perform this step.

To define the EA system in the PRD environment

1. Verify that the @JOB0120 job card information is correct.
2. Submit job @JOB0120 for execution.

   The SCL syntax required by this job is located in the *uprfx*.RI.JCLLIB library, member name SCLDEF2. This job defines the types, processor groups, and processor symbolics for the EA system in the PRD environment.
3. View the C1MSGS1 and C1MSGS2 reports produced by this job to verify that the actions were successfully executed.

Example: C1MSGS2 Summary Report for @JOB0120

```
Action                      Object          RC Location
--------------------------- --------------- -- ----------
DEFINE TYPE                 ASMPGM          0  To Env PRD
DEFINE PROCESSOR GROUP      *NOPROC*        0  To Env PRD
DEFINE PROCESSOR GROUP      LINKRN          0  To Env PRD
DEFINE PROCESSOR SYMBOL                     0  To Env PRD
DEFINE TYPE                 ASMPGM          0  To Env PRD
DEFINE PROCESSOR GROUP      *NOPROC*        0  To Env PRD
DEFINE PROCESSOR GROUP      LINKRN          0  To Env PRD
DEFINE PROCESSOR SYMBOL                     0  To Env PRD
DEFINE TYPE                 PARM            0  To Env PRD
DEFINE PROCESSOR GROUP      *NOPROC*        0  To Env PRD
DEFINE TYPE                 PARM            0  To Env PRD
DEFINE PROCESSOR GROUP      *NOPROC*        0  To Env PRD
```

13-Add the CA Endevor SCM Tables to the EA System

Job @JOB0130 adds all the CA Endevor SCM tables into the PRD, EA system. The CA Endevor SCM software is executed to perform this step.

To add the CA Endevor SCM tables into the EA system

1. Verify that the @JOB0130 job card information is correct.
2. Submit @JOB0130 for execution.

   The SCL syntax required by this job is located in the *uprfx*.RI.JCLLIB library, member name SCLACT2. The table source members required by this job are located in the *uprfx*.RI.JCLLIB library. Submit job @JOB0130 for execution. This job adds all the CA Endevor SCM tables into the PRD, EA system.
3. View the C1MSGS1 and C1MSGS2 reports produced by this job to verify that the actions were successfully executed.

Example: C1MSGS2 Summary Report for @JOB0130

```
                      PROC  NDVR   +--FROM  INFORMATION --+
ACTION    ELEMENT      RC    RC
ADD       BC1TNEQU    0000  0000   DSNAME=uprfx.RI.JCLLIB
ADD       C1DEFLTS    0000  0000   DSNAME=uprfx.RI.JCLLIB
ADD       C1UEXITS    0000  0000   DSNAME=uprfx.RI.JCLLIB
ADD       ENCOPTBL    0000  0000   DSNAME=uprfx.RI.JCLLIB
ADD       ENDICNFG    0000  0000   DSNAME=uprfx.RI.JCLLIB
ADD       ESYMBOLS    0000  0000   DSNAME=uprfx.RI.JCLLIB
ADD       ESMTPTBL    0000  0000   DSNAME=uprfx.RI.JCLLIB
ADD       GLBLTYPE    0000  0000   DSNAME=uprfx.RI.JCLLIB
```

14-Promote the CA Endevor SCM Tables and Processors

Job @JOB0140 issues a MOVE action to move the tables and processors added in @JOB0110 and @JOB0130 from the PRD, EA stage 1 location to the PRD, EA stage 2 location. The CA Endevor SCM software is executed to perform this step.

To promote the CA Endevor SCM tables and processors

1. Verify that the @JOB0140 job card information is correct.
2. Submit job @JOB0140 for execution.

   The SCL syntax required by this job is located in the *uprfx*.RI.JCLLIB library, member name SCLACT3. This job moves the tables and processors added in @JOB0110 and @JOB0130 from the PRD, EA stage 1 location to the PRD, EA stage 2 location.
3. View the C1MSGS1 and C1MSGS2 reports produced by this job to verify that the actions were successfully executed.

Example: C1MSGS2 Summary Report for @JOB0140

ACTION	ELEMENT	PROC RC	NDVR RC	SUBSYSTEM	TYPE
MOVE	DELACMOP	0000	0000	PROCESS	PROCESS
MOVE	DPDSMBR	0000	0000	PROCESS	PROCESS
MOVE	DSANDBOX	0000	0000	PROCESS	PROCESS
MOVE	GASM	0000	0000	PROCESS	PROCESS
MOVE	GASMMAP	0000	0000	PROCESS	PROCESS
MOVE	GCOB	0000	0000	PROCESS	PROCESS
MOVE	GLINK	0000	0000	PROCESS	PROCESS
MOVE	GPDSMBR	0000	0000	PROCESS	PROCESS
MOVE	GSANDBOX	0000	0000	PROCESS	PROCESS
MOVE	MPDSMBR	0000	0000	PROCESS	PROCESS
MOVE	BC1TNEQU	0000	0000	TABLES	ASMPGM
MOVE	C1DEFLTS	0000	0000	TABLES	ASMPGM
MOVE	C1UEXITS	0000	0000	TABLES	ASMPGM
MOVE	ENCOPTBL	0000	0000	TABLES	ASMPGM
MOVE	ENDICNFG	0000	0000	TABLES	ASMPGM
MOVE	ESMTPTBL	0000	0000	TABLES	ASMPGM
MOVE	ESYMBOLS	0000	0000	TABLES	ASMPGM
MOVE	GLBLTYPE	0000	0000	TABLES	PARM

15-Copy the Table Load Modules to the CSIQAUTU Library

Job @JOB0145 copies the Rapid Implementation table load modules from the PRD environment, stage 2, EA system to the user CSIQAUTU (*iprfx.iqual.CSIQAUTU*) that was created as part of the CA Endevor SCM installation.

To copy the table load modules to the CSIQAUTU library

1. Verify that the @JOB0145 job card information is correct.

2. Submit job @JOB0145 for execution.

   The Rapid Implementation table load modules are copied from the PRD environment, stage 2, EA system to the CSIQAUTU.

   Note: Previous copies of table load modules are replaced.

3. View the job output to ensure the table load modules were successfully copied.

16-Define the APPL@SYS System in the PRD Environment

Job @JOB0150 defines the types, processor groups, and processor symbolics for the APPL@SYS system in the PRD environment. The CA Endevor SCM software is executed to perform this step.

To define the APPL@SYS system in the PRD environment

1. Verify that the @JOB0150 job card information is correct.
2. Submit job @JOB0150 for execution.

   The SCL syntax required by this job is located in the *uprfx*.RI.JCLLIB library, member name SCLDEF3. This job defines the types, processor groups, and processor symbolics for the APPL@SYS system in the PRD environment.
3. View the C1MSGS1 and C1MSGS2 reports produced by this job to verify that the actions were successfully executed.

Note: @JOB0150—@JOB0170 define the type related definitions for the APPL@SYS system and @JOB0180-@JOB0200 define the same definitions for the COMM@SYS system. Due to the amount of data in the C1MSGS2 report and the fact that these jobs are likely to be enhanced, a copy of the C1MSGS2 report is not shown in this document.

17-Define the APPL@SYS System in the QA Environment

Job @JOB0160 defines the types, processor groups, and processor symbolics for the APPL@SYS system in the QA environment. The CA Endevor SCM software is executed to perform this step.

To define the APPL@SYS system in the QA Environment

1. Verify that the @JOB0160 job card information is correct.
2. Submit job @JOB0160 for execution.

   The SCL syntax required by this job is located in the *uprfx*.RI.JCLLIB library, member name SCLDEF4. This job defines the types, processor groups, and processor symbolics for the APPL@SYS system in the QA environment.
3. View the C1MSGS1 and C1MSGS2 reports produced by this job to verify that the actions were successfully executed.

18-Define the APPL@SYS System in the DEV Environment

Job @JOB0170 defines the types, processor groups, and processor symbolics for the APPL@SYS system in the DEV environment. The CA Endevor SCM software is executed to perform this step.

To define the APPL@SYS system in the DEV environment

1. Verify that the @JOB0170 job card information is correct.
2. Submit job @JOB0170 for execution.

   The SCL syntax required by this job is located in the *uprfx*.RI.JCLLIB library, member name SCLDEF5. This job defines the types, processor groups, and processor symbolics for the APPL@SYS system in the DEV environment.
3. View the C1MSGS1 and C1MSGS2 reports produced by this job to verify that the actions were successfully executed.

19-Define the COMM@SYS System in the PRD Environment

Job @JOB0180 defines the types, processor groups, and processor symbolics for the COMM@SYS system in the PRD environment. The CA Endevor SCM software is executed to perform this step.

To define the COMM@SYS system in the PRD environment

1. Verify that the @JOB0180 job card information is correct.
2. Submit job @JOB0180 for execution.

   The SCL syntax required by this job is located in the *uprfx*.RI.JCLLIB library, member name SCLDEF6. This job defines the types, processor groups, and processor symbolics for the COMM@SYS system in the PRD environment.
3. View the C1MSGS1 and C1MSGS2 reports produced by this job to verify that the actions were successfully executed.

20-Define the COMM@SYS System in the QA Environment

Job @JOB0190 defines the types, processor groups, and processor symbolics for the COMM@SYS system in the QA environment. The CA Endevor SCM software is executed to perform this step.

To define the COMM@SYS system in the QA environment

1. Verify that the @JOB0190 job card information is correct.
2. Submit job @JOB0190 for execution.

   The SCL syntax required by this job is located in the *uprfx*.RI.JCLLIB library, member name SCLDEF7. This job defines the COMM@SYS System in the QA Environment.
3. View the C1MSGS1 and C1MSGS2 reports produced by this job to verify that the actions were successfully executed.

21-Define the COMM@SYS System in the DEV Environment

Job @JOB0200 defines the types, processor groups, and processor symbolics for the COMM@SYS system in the DEV environment. The CA Endevor SCM software is executed to perform this step.

To define the COMM@SYS system in the DEV environment

1. Verify that the @JOB0200 job card information is correct.
2. Submit job @JOB0200 for execution.

   The SCL syntax required by this job is located in the *uprfx*.RI.JCLLIB library, member name SCLDEF8. This job defines the types, processor groups, and processor symbolics for the COMM@SYS system in the DEV environment.
3. View the C1MSGS1 and C1MSGS2 reports produced by this job to verify that the actions were successfully executed.

Optional Jobs to Delete Rapid Implementation Files

If you need to restart the Rapid Implementation process, you must first delete the files created by the Rapid Implementation jobs you have already run. Optional JCL jobs are supplied in the Rapid Implementation data set that you can use to delete Rapid Implementation files.

Delete Rapid Implementation Files

Important! The jobs in this section should only be executed if you need to restart the Rapid Implementation and therefore need to delete previously allocated files.

This task is optional and removes the files allocated by the jobs run during the Rapid Implementation.

To delete Rapid Implementation files

1. Verify the job card and submit @JOBDEL for execution.

   This job deletes all files previously allocated by jobs @JOB0070, @JOB0080, @JOB2070 and @JOB2200. These files are the base libraries for all the APPL@SYS, COMM@SYS, and APPL@SYS2 systems.
2. Verify the job card and submit @JOBDLEA for execution.

   This job deletes all files for the EA system previously allocated by @JOB0010.
3. Verify the job card and submit @JOBDLVS for execution.

   This job deletes all VSAM files previously allocated.

How to Verify the Implementation

You need to verify that the implementation you set up in the previous steps is properly established and that it is able to accept inventory. The supplied @JOB10*nn* series of jobs takes you step-by-step through the process of verifying your implementation.

1. Add elements of various types into the DEV environment.
2. Create and cast a promotion package.
3. Execute the promotion package until the inventory reaches the final inventory location, PRD stage 2.
4. Verify that the inventory exists in the final inventory location and that the elements and package can be browsed.
5. Delete all the verification data.

1-Add Test Elements to the DEV Environment

Job @JOB1000 executes CA Endevor SCM to add elements into the DEV stage 1 environment.

To add test elements to the DEV environment

1. Verify that the @JOB1000 job card information is correct.
2. Submit job @JOB1000 for execution.

   The SCL syntax required by this job is located in the *uprfx*.RI.JCLLIB library, member name SCLACT4. The SCL is set up to obtain the element source from the *iprfx.iqual.CSIQSAMP* library, which was delivered as part of the CA Endevor SCM installation. Job @@JOB1000 adds elements to the DEV environment.
3. View the C1MSGS1 and C1MSGS2 reports produced by this job to verify that the elements actions were successfully executed.

Example: C1MSGS2 Sample Report for @JOB1000

```
                        PROC  NDVR   +---FROM  INVENTORY LOC ---+
ACTION     ELEMENT       RC    RC    ENVIRONMENT   SYSTEM
ADD        FAPHDR              0000  DSNAME=iprfx.iqual.CSIQSAMP
ADD        FAPPAGE             0000  DSNAME=iprfx.iqual.CSIQSAMP
ADD        FAPASM01    0000    0000  DSNAME=iprfx.iqual.CSIQSAMP
ADD        FAPCOB01    0000    0000  DSNAME=iprfx.iqual.CSIQSAMP
```

2-Delete, Define, Cast a Test Promotion Package

Job @JOB1010 executes CA Endevor SCM to delete, define and cast a promotion package.

To delete, define, and cast a test promotion package

1. Verify that the @JOB01010 job card information is correct.
2. Submit job @JOB01010 for execution.

   The SCL syntax required by this job is located in the *uprfx*.RI.JCLLIB library, member names SCLPKG1 and SCLACT5. The test package is deleted, defined, and then cast.
3. View the C1MSGS1 and C1MSGS2 reports produced by this job to verify that the package actions were successfully executed.

   This job is set up as part of a repeatable process; therefore the delete action will usually complete with a return code of 4, because initially the package does not exist.

Example: C1MSGS2 Summary Report for @JOB1010

```
Action       Statement  Package  Package          Return
Name          Number    Number   Name              Code
-----------  ---------  -------  ----------------  -------
DELETE             1         1   PKGRI001             4
DEFINE             2         1   PKGRI001             0
CAST               3         1   PKGRI001             0
```

We recommend that you execute CA Endevor SCM at this time to view the inventory that was added in the previous step and to view the package that was created in this step. For more information about starting CA Endevor SCM in foreground, see the topics Installation Considerations and How to Define CA Endevor SCM to the ISPF in the chapter "Installing CA Endevor SCM" in the *Installation Guide*.

Try several of the CA Endevor SCM element features using Quick Edit such as building a list of elements, displaying the master information, and browsing the element. Also try several of the package display features such as displaying a cast report and displaying SCL.

3-Execute the Test Promotion Package

Job @JOB1020 executes CA Endevor SCM to execute the test promotion package to promote the test elements to the DEV Stage 2 location.

To execute the test promotion package

1. Verify that the @JOB1020 job card information is correct.
2. Submit job @JOB1020 for execution.

   The SCL syntax required by this job is located in the *uprfx*.RI.JCLLIB library, member name SCLPKG2. The test promotion package is executed.
3. View the C1MSGS1 and C1MSGS2 reports produced by this job to verify that the elements actions were successfully executed.

Example: C1MSGS2 Summary Report for @JOB1020

```
Action       Statement  Package  Package          Return
Name          Number    Number   Name              Code
-----------  ---------  -------  ---------------  -------
EXECUTE            1          1  PKGRI001             0
```

4-Execute the Test Promotion Package Multiple Times

Job @JOB1030 executes CA Endevor SCM to execute the test promotion package multiple times to move test elements through the lifecycle to the final stage, PRD stage 2.

To execute the test promotion package multiple times

1. Verify that the @JOB01030 job card information is correct.
2. Submit job @JOB01030 for execution.

   The SCL syntax required by this job is located in the *uprfx*.RI.JCLLIB library, member name SCLPKG3. This job promotes the inventory to the final location in the map, which is PRD Stage 2.
3. View the C1MSGS1 and C1MSGS2 reports produced by this job to verify that the package actions were successfully executed.

Example: C1MSGS2 Summary Report for @JOB1030

```
Action       Statement  Package  Package           Return
Name          Number    Number   Name               Code
-----------  ---------  -------  ----------------  -------
EXECUTE            1         1   PKGRI001             0
EXECUTE            2         1   PKGRI001             0
EXECUTE            3         1   PKGRI001             0
```

We recommend that you execute CA Endevor SCM at this time to view the inventory that was previously added in and to view the promotion package that was executed multiple times in this step. Try several of the CA Endevor SCM element features using CA Endevor Quick Edit such as building a list of elements, displaying the master information and browsing the elements. Also, try several of the package display features such as displaying a cast report and displaying promotion history.

5-Delete Verification Data

Job @JOB1040 executes CA Endevor SCM to delete the test promotion package and delete the test elements associated with that package.

Important! You may want to execute this job at this time or you may want to execute this job at a later date. The purpose of this job is to remove the test data. It will not cause any problems if this job is not executed.

To delete verification data

1. Verify that the @JOB01040 job card information is correct.
2. Submit job @JOB01040 for execution.

   The SCL syntax required by this job is located in the *uprfx*.RI.JCLLIB library, member names SCLPKG4 and SCLACT6. This job deletes the promotion package and deletes the elements used by that package.
3. View the C1MSGS1 and C1MSGS2 reports produced by this job to verify that the package actions were successfully executed.

Example: C1MSGS2 Summary Report for @JOB1040

```
Action       Statement  Package  Package           Return
Name          Number    Number   Name               Code
-----------  ---------  -------  ----------------  -------
DELETE             1          1   PKGRI001             0

                      PROC  NDVR
ACTION    ELEMENT      RC    RC
DELETE    FAPHDR            0000
DELETE    FAPPAGE           0000
DELETE    FAPASM01    0000  0000
DELETE    FAPCOB01    0000  0000
```

How to Complete Your Basic Implementation

After you have verified that the lifecycle and inventory structure are set up correctly, you can begin to load your own inventory. You may also decide to add your own system and set up sandbox subsystems so that your developers can use different sandboxes for concurrent development purposes. To complete your basic setup, do the following:

1. Load your pilot application into the newly created environment.
2. (Optional) Create your own system.
3. (Optional) Define sandbox subsystems for concurrent development.

1-Load Your Inventory

After you have verified that the Rapid Implementation lifecycle and inventory structure are correctly set up, you can load your inventory into the inventory structure to put it under the control of CA Endevor SCM. You will need to understand how your inventory is configured and you may need to modify the names of your production JCL.

1. Determine where your inventory is physically located and how it is configured. You can do this using the Inventory Analyzer.
2. Load your inventory into the Rapid Implementation inventory structure. You can use the Load utility with the SCL generated by the Inventory Analyzer to add your inventory members. Alternately, you can use the CA Endevor SCM online batch panels to generate and submit SCL to load the inventory.

   Note: For more information about the Load utility, see the *Utilities Guide*. For more information about inventory structure, see the *Administration Guide*. For more information about the batch panels, see the *User Guide*.

Inventory Analyzer

The Inventory Analyzer is a tool that automatically categorizes your existing software components by examining component source and executables to determine component types.

The Inventory Analyzer automatically classifies existing software components into CA Endevor SCM types and processor groups. Types and processor groups are physical classification of components, such as JCL or COBOL for DB2.

The Inventory Analyzer provides a standard set of types and processor groups. If you want or need to, you can tailor type and processor group names to meet the inventory naming conventions you want to use.

Inventory Analyzer produces several reports that allow you to review selected portions of the analysis. In addition, the Inventory Analyzer creates SCL that you can use to load the elements into the logical inventory structure using the Load utility.

Note: For more information about the Inventory Analyzer, see the *Inventory Analyzer Guide*.

Plan to Modify Your Production JCL

One of the steps for the Rapid Implementation consists of adding your production JCL members under the control of CA Endevor SCM. You may need to develop a plan to change the name of the libraries specified in these JCL members to point to the new CA Endevor SCM controlled libraries. For example, the load library names.

2-Create an Additional System

This step is optional. You can choose to create an additional system in the Rapid Implementation inventory structure using jobs supplied by CA.

To create an additional system

1. Define the base and delta data sets for the DEV, QA, and PRD environments for the APPL@SY2 system using the following steps:
   a. Verify that the @JOB2070 job card information is correct.
   b. Submit job @JOB2070 for execution.

      The base and delta data sets for the DEV, QA, and PRD environments are defined for the APPL@SY2 system. One base library is defined for each of the types and one delta library is defined to support the system/subsystem. IEFBR14 is used to define these data sets. The data set attributes vary, depending on the type of data the data set will contain.
2. Define the types, processor groups, and processor symbolics for the APPL@SY2 system in the PRD environment using the following steps:
   a. Verify that the @JOB2150 job card information is correct.
   b. Submit job @JOB2150 for execution.

      The SCL syntax required by this job is located in the *uprfx*.RI.JCLLIB library, member name SCLDE23. The CA Endevor SCM software is executed to perform this step. The types, processor groups, and processor symbolics are defined for the APPL@SY2 system in the PRD environment.
   c. View the C1MSGS1 and C1MSGS2 reports produced by this job to verify that the actions were successfully executed.
3. Define the types, processor groups, and processor symbolics for the APPL@SY2 system in the QA environment using the following steps:
   a. Verify that the @JOB2160 job card information is correct.
   b. Submit job @JOB2160 for execution.

      The SCL syntax required by this job is located in the *uprfx*.RI.JCLLIB library, member name SCLDE24. The CA Endevor SCM software is executed to perform this step. The types, processor groups, and processor symbolics are defined for the APPL@SY2 system in the QA environment.
   c. View the C1MSGS1 and C1MSGS2 reports produced by this job to verify that the actions were successfully executed.
4. Define the types, processor groups, and processor symbolics for the APPL@SY2 system in the DEV environment.
   a. Verify that the @JOB2170 job card information is correct.
   b. Submit job @JOB2170 for execution.

The SCL syntax required by this job is located in the *uprfx*.RI.JCLLIB library, member name SCLDE25. The CA Endevor SCM software is executed to perform this step. The types, processor groups, and processor symbolics are defined for the APPL@SY2 system in the DEV environment.

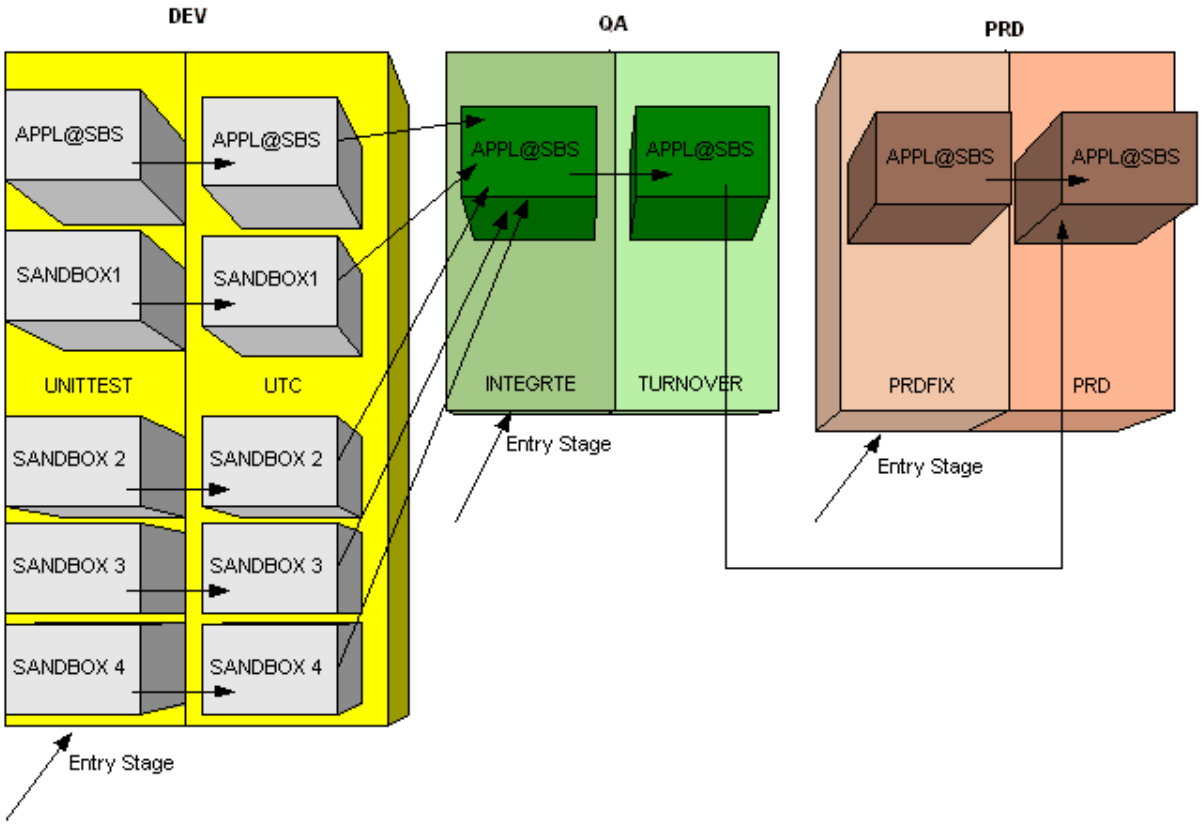
c. View the C1MSGS1 and C1MSGS2 reports produced by this job to verify that the actions were successfully executed.

3-Sandbox Development

Sandbox development allows more than one person to work on the same software at the same time. Sometimes this is a result of parallel projects that require multiple resources, but the more common condition is a project or change that is in a development phase, and the current version running in production requires an immediate change to address an abend situation.

A *sandbox* is a segregated private work area that is a subsystem.

The following diagram shows the sandbox lifecycle you can set up using Rapid Implementation. Sandboxes allow developers to work in the DEV environment without affecting other users. Changes made in these subsystems are integrated at stage 1 of the QA environment. The arrows between the stages indicate how the stages are mapped to move your software inventory from stage to stage.



How to Define Sandbox Subsystems

Defining sandboxes is optional. Sandboxes are intended to be used for concurrent development. This type of subsystem allows changes to be made in the DEV environment without affecting other users. Changes made in these subsystems are integrated at stage 1 of the QA environment. Typically, you will want to create several sandbox subsystems.

Two methods of defining sandboxes are recommended. Use one of the following methods:

- Use the supplied jobs @JOB2200 and @JOB2210. For more information, see Defining Sandboxes Using Supplied Jobs (see page 164).
- Use the supplied GSANDBOX processor. For more information, see Defining Sandboxes Using Supplied Processor GSANDBOX (see page 165). Sandboxes created using this processor can be easily deleted using the DSANDBOX processor. For more information, see Delete Sandboxes Added by GSANDBOX (see page 166).

Define Sandboxes Using Supplied Jobs

You can use the the Rapid Implementation sandbox jobs to create sandboxes. Additional sandboxes can be set up at a future date.

To define sandbox subsystems using supplied jobs

1. Define the sandbox subsystem files using the following steps:
   a. If you are only defining one sandbox, you can skip to (b.). If more than one sandbox is required, you must edit the @JOB2200 to add additional JCL EXEC statements for each additional sandbox you plan to create.
      - Edit the @JOB2200 member.
      - A pair of EXEC statements (STEP0010 and STEP0020) exists to create one sandbox subsystem at the DEV stage 1 and DEV stage 2 locations.
      - Repeat this pair of EXEC statements for each additional sandbox you plan to create. Increment the step names and change the "SUB=" parameters to the name of the additional sandbox. You can use any valid subsystem name. If a sandbox is to be used by a particular developer, then we recommend you use the developer's user ID as the name. If multiple developers are sharing a sandbox, then name the sandbox according to the project, feature or CCID that the group is working on.
      - Repeat this process for each additional sandbox.
   b. Verify the @JOB2200 job card information is correct.
   c. Submit job @JOB2200 for execution.

   This job allocates files for the SAND@BOX subsystems in the DEV environment, APPL@SYS system.

2. Define the SAND@BOX subsystems in the DEV environment following these steps:

   a. If you are only defining one sandbox, you can skip to (b.). If you modified @JOB2200 to create more than one sandbox, you must edit the SCL SCLDE26 member to add additional SCL statements for each additional sandbox you plan to create.

      - Edit the SCLDE26 member located in the *uprfx.*RI.JCLLIB library.
      - Repeat the entire SCL DEFINE statement including the ending period. Change the subsystem name to the additional sandbox subsystem.
      - Repeat this process for each additional sandbox subsystem.

   b. Verify the @JOB2210 job card information is correct.

   c. Submit job @JOB2210 for execution.

   This defines the SAND@BOX subsystems in the DEV environment, APPL@SYS system. These subsystems map to the APPL@SBS subsystem in the QA environment, which is the next environment up the map.

Define Sandboxes Using Supplied Processor GSANDBOX

You can use the supplied sandbox generate processor, GSANDBOX, to create sandbox subsystems.

To define sandbox subsystems using supplied processor simply add an element of type SANDBOX to the DEV environment of the APPL@SBS subsystem in APPL@SYS system. This invokes the GSANDBOX processor. The content of the element is immaterial, only the element name is used in the processor and that is used as the new Sandbox subsystem name. If you want to, you can supply comments or notes in the element.

The GSANDBOX processor allocates the libraries required for the new subsystem and defines the sandbox subsystem in the DEV environment, APPL@SYS system. This subsystem maps to the APPL@SBS subsystem in the QA environment, which is the next environment up the map.

As delivered, the processor allocates the same list of base, delta and processor output files with the same allocations that are used for the normal APPL@SBS subsystem (except for the SANDBOX base library). If any adjustments were made to the APPL@SBS allocations, the same adjustments need to be made to the GSANDBOX processor.

GSANDBOX first attempts to list all of the elements in the subsystem that you are trying to create. This step should fail (the subsystem should not already exist); if it does the processor allocates the subsystem files and submits a job to the internal reader to define the subsystem. If the "list element" step does not fail (return code less than 12), the processor ends with an error.

Delete Sandboxes Added by GSANDBOX

The Sandbox delete processor, DSANDBOX, is invoked when an element of type SANDBOX is deleted. This processor deletes the sandbox subsystem definition and deletes files allocated by GSANDBOX for the subsystem.

As delivered, this processor deletes the same list of base, delta and processor output files with the same allocations that are used for the normal APPL@SBS subsystem (except for the SANDBOX base library). Any adjustments to the APPL@SBS allocations should be made to the GSANDBOX and DSANDBOX processors as well if they are going to be used to create and delete sandbox subsystems.

DSANDBOX attempts to list all of the elements in stage 1 and stage 2 of the sandbox subsystem. The result from both of these steps should be a return code 4, which indicates that no elements exist at either stage. If this is the case, the files are deleted and a job is submitted to the internal reader to delete the subsystem definition. If elements do exist in the sandbox subsystem, the processor ends with an error.

JCL Library Contents

All of the members delivered in the Rapid Implementation data sets are described next. When you run the primary implementation job @@UPDJCL, that job updates all of the other members with your site–specific parameters.

Primary Implementation Jobs

@@UPDJCL

JCL that updates all members of the JCL data set with parameters specified by the client for data set name prefixes as well as CA Endevor SCM parameters and definitions.

@INDEX

A list of members that reside in the Rapid Implementation JCL data set. Exists as reference material.

@JOB0010

JCL that deletes and defines all the data sets required to set up the EA Administrative System.

@JOB0020

JCL that assembles and link edits the C1DEFLTS and ESYMBOLS tables. Add the GLBLTYPE global type sequence table to the PARMLIB.

@JOB0030

JCL that deletes and defines ACM Query VSAM files.

@JOB0040

JCL that deletes and defines package VSAM file.

@JOB0050

JCL that deletes and defines element catalog and element index VSAM files.

@JOB0060

JCL that deletes and defines MCF VSAM files for environment setup.

@JOB0070

Deletes and defines base libraries for the APPL@SYS DEV, QA, and PRD environment.

@JOB0080

Deletes and defines base libraries for the COMM@SYS DEV, QA, and PRD environment.

@JOB0100

Defines the EA system.

@JOB0110

Adds processors to the EA system.

@JOB0120

Defines types, processor groups, and processor symbolics to the EA system.

@JOB0130

Adds the CA Endevor SCM tables into the EA system.

@JOB0140

Promotes the CA Endevor SCM tables in the EA system.

@JOB0145

Copies table load modules to the user CSIQAUTU library.

@JOB0150

Defines the APPL@SYS system to the PRD environment.

@JOB0160

Defines the APPL@SYS system to the QA environment.

@JOB0170

Defines the APPL@SYS system to the DEV environment.

@JOB0180

Defines the COMM@SYS system to the PRD environment.

@JOB0190

Defines the COMM@SYS system to the QA environment.

@JOB0200

Defines the COMM@SYS system to the DEV environment.

Implementation Verification Jobs

@JOB1000

Adds test elements to the APPL@SYS system to test setup and configuration.

@JOB1010

Deletes defines, and casts a promotion package.

@JOB1020

Executes the promotion package.

@JOB1030

Executes the promotion package multiple times to move test elements through the lifecycle to the final location in the map, PRD stage 2.

@JOB1040

Deletes the test elements from the final stage in the lifecycle and delete the test package.

Optional Implementation Jobs

@JOB2070

Deletes and defines base libraries for the APPL@SY2 DEV, QA and PRD environment.

@JOB2150

Defines the APPL@SY2 system to the PRD environment.

@JOB2160

Defines the APPL@SY2 system to the QA environment.

@JOB2170

Defines the APPL@SY2 system to the DEV environment.

@JOB2200

Allocates SAND@BOX files for setting up SAND@BOX subsystems in the DEV environment.

@JOB2210

Defines SAND@BOX subsystems. Note that the NEXT SUBSYSTEM in the DEFINE SUBSYSTEM SCL statement is APPL@SBS.

Post-Implementation Sample Jobs

Note: For more information about the CA Endevor SCM utilities used in these jobs, see the *Utilities Guide*.

@JOB9000

Establishes GDG bases for VSAM backups and CA Endevor SCM incremental backups.

@JOB9010

Reorgs and backs up VSAM files.

@JOB9020

Executes CA Endevor SCM full unload.

@JOB9030

Executes CA Endevor SCM incremental unload.

@JOB9040

Executes CA Endevor SCM reload and create generate SCL.

@JOB9050

Executes generate SCL created in @JOB9040.

@JOB9060

Executes the CA Endevor SCM analyzer utility.

@JOB9070

Executes the CA Endevor SCM load utility.

@JOB9080

Builds SCL for all environments.

@JOB9090

Builds SCL for the APPL@SYS system for all environments.

CA Endevor SCM Tables Modified by the Rapid Implementation Process

BC1TNEQU

Contains the CA Endevor SCM external security interface table.

C1DEFLTS

Contains CA Endevor SCM defaults table that defines three environments. This table is assembled and link edited in @JOB0020. This table is added into your EA system in @JOB0130.

C1UEXITS

Contains the CA Endevor SCM exits table.

ENCOPTBL

Contains the CA Endevor SCM optional features table. This table is added into your EA system in @JOB0130.

ENDICNFG

Contains the CA Endevor SCM ISPF dialog configuration table. This table is added into your EA system in @JOB0130.

ESMTPTBL

Contains the CA Endevor SCM SMTP email table. This table is added into your EA system in @JOB0130.

ESYMBOLS

Contains the CA Endevor SCM site-wide symbolics table. This is assembled and link edited in @JOB0020. This table is added into your EA system in @JOB0130.

GLBLTYPE

Contains the global type sequence table. This table is copied into the PARMLIB in @JOB0020. This table is added into your EA system in @JOB0130.

SCL and VSAM Definitions Referred to in the Rapid Implementation Jobs

SCLACT1

Contains SCL that adds processors to the EA system. Referred to in @JOB0110.

SCLACT2

Contains SCL that adds tables to the EA system. Referred to in @JOB0130.

SCLACT3

Contains SCL that moves all EA system elements from stage 1 to stage 2. Referred to in @JOB0140.

SCLACT4

Contains SCL that adds verification elements. Referred to in @JOB1000.

SCLACT5

Contains SCL that moves elements for the verification package. Referred to in @JOB1010.

SCLACT6

SCL to delete the verification elements. Referred to in @JOB1040.

SCLDEF1

SCL to define EA system. Referred to in @JOB0100.

SCLDEF2

SCL to define types to EA system. Referred to in @JOB0120.

SCLDEF3

SCL to define APPL@SYS system to PRD environment. Referred to in @JOB0150.

SCLDEF4

SCL to define APPL@SYS system to QA environment. Referred to in @JOB0160.

SCLDEF5

SCL to define APPL@SYS to DEV environment. Referred to in @JOB0170.

SCLDEF6

SCL to define COMM@SYS system to PRD environment. Referred to in @JOB0180.

SCLDEF7

SCL to define COMM@SYS system to QA environment. Referred to in @JOB0190.

SCLDEF8

SCL to define COMM@SYS system to DEV environment. Referred to in @JOB0200.

SCLDE23

SCL to define second system and subsystem to PRD environment. Referred to in @JOB2150.

SCLDE24

SCL to define second system and subsystem to QA environment. Referred to in @JOB2160.

SCLDE25

SCL to define second system and subsystem to DEV environment. Referred to in @JOB2170.

SCLDE26

SCL to define a SAND@BOX subsystem to the DEV environment. Referred to in @JOB2210.

SCLPKG1

SCL to delete, define and cast the verification promotion package. Referred to in @JOB1010.

SCLPKG2

SCL to execute the verification promotion package once. Referred to in @JOB1020.

SCLPKG3

SCL to execute the verification promotion package multiple times to move test elements through the lifecycle to the final stage, PRD stage 2. Referred to in @JOB1030.

SCLPKG4

SCL to delete the verification promotion package. Referred to in @JOB1040.

VSDACMR

Delete and define for ACM root VSAM file. Referred to in @JOB0030.

VSDACMX

Delete and define for ACM cross-reference VSAM file. Referred to in @JOB0030.

VSDEINDX

Delete and define for element index VSAM file. Referred to in @JOB0050.

VSDELMC

Delete and define for element catalog VSAM file. Referred to in @JOB0050.

VSDMCFD1

Delete and define for DEV stage 1 MCF VSAM file. Referred to in @JOB0060.

VSDMCFD2

Delete and define for DEV stage 2 MCF VSAM file. Referred to in @JOB0060.

VSDMCFP1

Delete and define for PRD stage 1 MCF VSAM file. Referred to in @JOB0060.

VSDMCFP2

Delete and define for PRD stage 2 MCF VSAM file. Referred to in @JOB0060.

VSDMCFQ1

Delete and define for QA stage 1 MCF VSAM file. Referred to in @JOB0060.

VSDMCFQ2

Delete and define for QA stage 2 MCF VSAM file. Referred to in @JOB0060.

VSDPKG

Delete and define for package VSAM file. Referred to in @JOB0040.

Other Members Included in the Rapid Implementation Data Set

@JOBDEL

Delete all files created by jobs @JOB0070, @JOB0080, @JOB2070 and @JOB2200.

Important! @JOBDEL should only be executed if you want to restart the Rapid Implementation and need to delete previously allocated files.

@JOBDLEA

Delete all files for the EA system created by @JOB0010.

Important! @JOBDLEA should only be executed if you want to restart the Rapid Implementation and need to delete previously allocated files.

@JOBDLVS

Delete all VSAM files created.

Important! @JOBDLVS should only be executed if you want to restart the Rapid Implementation and need to delete previously allocated files.

SCMMALEA

JCL procedure used to delete and allocate EA data sets.

SCMMALLO

JCL procedure used by @JOB0070, @JOB0080, @JOB2070 and @JOB2200 to allocate files for a single stage.

SCMMPROC

JCL procedure for executing all CA Endevor SCM steps in the @JOB0100 through @JOB0200 job streams.

Supplied Types, Processors, and Processor Groups

The predefined environment set up by the Rapid Implementation process includes predefined types, processor groups, and processors.

Types

The following types are included and defined to the APPL@SYS system, unless otherwise specified:

ASMPGM

Assembler programs (defined to both EA and APPL@SYS systems).

ASMMAC

Assembler macros.

CICS

Programs and maps.

CLSTREXX

TSO commands, CLIST and REXX procedures.

COBOL

COBOL programs.

COBCOPY

COBOL copybooks.

DB2

Binds and programs.

JCL

Job streams.

LNK

Program link decks.

PARM

PARMLIB members (EA system only).

PROCESS

CA Endevor SCM Processors (EA system only).

SANDBOX

Sandbox subsystem type.

Processors

The following processors are included:

- DELACMOP-Delete
- DPDSMBR-Delete
- GASM-Generate
- GASMMAP-Generate
- GCOB-Generate
- GLINK-Generate
- GPDSMBR-Generate
- MPDSMBR-Move

- DSANDBOX-Delete
- GSANDBOX-Generate

Processor Groups

The following processor groups are included:

NOPROC

For elements not requiring processors.

PROCESS

Assemble and link edit processors. Type PROCESS, system EA only.

- Generate processor GPPROCSS
- Move processor GPPROCSS
- Delete processor DPPROCSS

LINKRN

Assemble and link edit CA Endevor SCM tables, EA system only.

- Generate processor GASM
- Move processor *NOPROC*
- Delete processor DPDSMBR

LOAD

Assemble and link edit.

- LET NORENT NOREUS
- LET RENT REUS (GCOB processor only)

Generate processor.

- GASM (Type ASMPGM)
- GLINK (Type LNK)
- GCOB (Type COBOL)

Move processor-MPDSMBR

Delete processor-DPDSMBR

LOADRN

Assemble and link edit.

- NOLET RENT REUS

Generate processor.

- GASM (Type ASMPGM)
- GLINK (Type LNK)

Move processor-MPDSMBR

Delete processor-DPDSMBR

LOADRNM

Assemble and link edit.

NOLET RENT REUS

Generate processor.

- GASM (Type ASMPGM)
- GLINK (Type LNK)

Move processor-MPDSMBR

Delete processor-DPDSMBR

LOADRU

Assemble and link edit.

NOLET NORENT REUS

Generate processor.

- GASM (Type ASMPGM)
- GLINK (Type LNK)

Move processor MPDSMBR

Delete processor DPDSMBR

MAPASM

Assemble and link edit CICS maps.

Generate processor-GASMMAP

Move processor-MPDSMBR

Delete processor-DELACMOP

OBJ

Assemble only.

NORENT NOREUS

Generate processor-GASM

Move processor-MPDSMBR

Delete processor-DPDSMBR

OBJRN

Assemble only.

RENT REUS

Generate processor-GASM

Move processor-MPDSMBR

Delete processor-DPDSMBR

SANDBOX

Allocate and define sandbox subsystem.

Generate processor-GSANDBOX

Delete processor-DSANDBOX

Move processor-MPDSMBR

Appendix C: CA Endevor Quick Edit

This section contains the following topics:

How to Implement CA Endevor Quick Edit

After CA Endevor SCM is installed, you can implement CA Endevor Quick Edit, using the following steps:

1. Update the CA Endevor SCM Defaults Table (C1DEFLTS).
2. Define CA Endevor Quick Edit to ISPF.
3. (Optional) Update ISPF panel default values for selected fields using ENDICNFG.
4. (Optional) Write the Edit Session Startup command.
5. Review ENDES000.
6. Review Exit support.

Update the CA Endevor Defaults Table (C1DEFLTS)

To enable CA Endevor Quick Edit, you must update, assemble and link-edit the CA Endevor SCM Defaults Table. Review and update the following parameters.

To update the CA Endevor SCM Defaults Table

1. Type **Y** as the value for the EDITELM parameter in the TYPE=MAIN macro:

   EDITELM=Y,

   This allows the CA Endevor Quick Edit Option to be enabled at your site.

2. (Optional) Review and optionally change the following parameters to your preferences. These options affect both standard CA Endevor SCM and CA Endevor Quick Edit.

   MIXEDFMT

   Specifies whether mixed-case entries are allowed in the CCID, COMMENT, and DESCRIPTION fields.

   SOFETCH

   Specifies whether an element that is fetched can be signed out to you, if it is not already signed out to someone else.

3. Reassemble and link-edit the table.

   The CA Endevor Quick Edit option is enabled. Also, any other changes you made to the table are enabled.

Note: For more information about the Defaults Table, see the *Administration Guide*.

Define CA Endevor Quick Edit to ISPF

You can invoke CA Endevor Quick Edit from any standard ISPF selection menu. Use the CLIST provided with the installation files, which is member ENDICLS1 in CSIQCLSO.

To define CA Endevor Quick Edit to ISPF:

1. Copy member ENDICLS1 in CSIQCLSO to a data set that is allocated to the standard SYSPROC DD concatenation. Edit the CLIST to meet the requirements of your site.

   Note: For more information, see the file ENDICLS1.

   The CLIST is ready for your site.

2. Change an ISPF primary or secondary options panel to include an option for CA Endevor Quick Edit.

   This enables the CLIST, so that when you select this option on ISPF, the CLIST is invoked to begin CA Endevor Quick Edit.

CA Endevor Quick Edit is enabled on the following sample panel. The first bold line indicates what option the user enters to select CA Endevor Quick Edit. The second bold line indicates that the CA Endevor Quick Edit CLIST ENDICLS1 should be executed when the user selects the defined option.

```
%---------------------- ISPF/PDF PRIMARY OPTION MENU ----------------------
%OPTION ===>_ZCMD +
%
       +USERID - &ZUSER
% 0 +ISPF PARMS - Specify terminal and user parameters           +TIME - &ZTIME
% 1 +BROWSE - Display source data or output listings
       +TERMINAL - &ZTERM
% 2 +EDIT - Create or change source data                                 +PF
KEYS - &ZKEYS
% 3 +UTILITIES - Perform utility functions
% 4 +FOREGROUND - Invoke language Processors in foreground
% 5 +BATCH - Submit job for language processing
% 6 +COMMAND - Enter TSO Command, CLIST, or REXX exec
% 7 +DIALOG TEST - Perform dialog testing
% 8 +LM UTILITIES- Perform library administrator utility functions
% 9 +IBM PRODUCTS- Additional IBM program development products
% 10 +SCLM - Software Configuration and Library Manager
% E +Endevor - Endevor Software Management System
% EE +EDIT ELEMENT- Endevor Quick Edit
% C +CHANGES - Display summary of changes for this release
% T +TUTORIAL - Display information about ISPF/PDF
% X +EXIT - Terminate ISPF using log and list defaults
%
+Enter%End+command to terminate ISPF.
)INIT
)PROC
&ZQ = &Z
  IF (&ZCMD ¬= ' ')
    &ZQ = TRUNC(&ZCMD,'.')
    IF (&ZQ = ' ')
      .MSG = ISRU000
  &ZSEL = TRANS( &ZQ
                0,'PANEL(ISPOPTA)'
                1,'PGM(ISRBRO) PARM(ISRBRO01)'
                2,'PGM(ISREDIT) PARM(P,ISREDM01)'
                3,'PANEL(ISRUTIL)'
                4,'PANEL(ISRFPA)'
                5,'PGM(ISRJB1) PARM(ISRJPA) NOCHECK'
                6,'PGM(ISRPTC)'
                7,'PGM(ISPYXDR) PARM(ISR) NOCHECK'
                8,'PANEL(ISRLPRIM)'
                9,'PANEL(ISRDIIS)'
               10,'PGM(ISRSCLM) NOCHECK'
                E,'CMD(Endevor)'
               EE,'CMD(ENDICLS1)'
                C,'PGM(ISPTUTOR) PARM(ISR00005)'
                T,'PGM(ISPTUTOR) PARM(ISR00000)'
              ' ',' '
                X,'EXIT'
                *,'?' )
  &ZTRAIL = .TRAIL
)End
```

Update ISPF Default Configuration Table Values

You can change the delivered default values specified in the ISPF Default Configuration table (ENDICNFG). The default values affect how CA Endevor SCM processes actions or affect the selection lists returned by CA Endevor SCM. The same ISPF Default Configuration table (ENDICNFG) is used by standard CA Endevor SCM and CA Endevor Quick Edit. However, some options are specific to CA Endevor Quick Edit.

For more information, see the chapter "Using the ISPF Dialog Options Configuration Table" in the *Administration Guide*.

The user can change some of the options in effect for their own user ID. For more information, see Change Action Options in the *CA Endevor Quick Edit Option User Guide*.

To update the ISPF Default Configuration table

1. Review the parameters in the ISPF Configuration table (ENDICNFG) and make any changes as appropriate for your site.

   a. The following parameters affect Generate processing in CA Endevor Quick Edit.

      Note: The user can change defaults in effect for their own user ID on the Actions Options panel.

      GENERATE_ELEMENT

      Specifies whether elements are automatically generated as part of CA Endevor Quick Edit action processing.

      Y – The element being acted upon is generated. This is the default.

      N – The element being acted upon is *not* generated.

      Note: This option also affects standard CA Endevor SCM transfer processing.

      QE_GENERATE_ IN_PLACE

      Specifies whether elements are generated in the Stage where found, or whether they are copied back to the entry Stage before being generated.

      Y – Elements are generated in the Stage where they are found.

      N – Elements are copied back to the entry Stage and then generated in the entry Stage. This is the default.

QE_NOSOURCE

Specifies whether the NOSOURCE option is used on the GENERATE action.

Y – When the source for the element that is the target of a Generate action is not found at the target location, the first occurrence of the element found up the map from the target location is used as the element's source and is input to the Generate Processor. However, the element's source is not fetched back to the target location from up the map. After the Generate action completes, the targeted location coded in a Generate action contains the outputs created by the Generate Processor.

N – When the source for the element that is the target of a Generate action is not found at the target location, the first occurrence of the element found up the map from the target location is copied back to the target location before being generated. This is the default.

GENERATE_MODE

Specifies whether elements are generated in Foreground or Batch.

F – Generate processing is performed in foreground. This is the default.

B – Generate processing is performed in batch.

Note: If GENERATE_MODE=F, then QE_AUTOGEN cannot be Y.

Note: This option also affects standard CA Endevor SCM Generate processing.

QE_AUTOGEN

Specifies that when an element is added, updated, or generated, then the elements that use this component element are automatically generated. The option only applies to these actions: ADD, UPDATE and GENERATE.

Note: If the value of GENERATE_MODE is F, QE_AUTOGEN cannot be Y.

Y – When an element is acted upon by the Add, Update, or Generate action, then all the elements that use this component element are automatically generated.

N – When an element is acted upon by the Add, Update, or Generate action, then all the elements that use this component element are *not* automatically generated. This is the default.

b. The following parameters affect Delete action processing in CA Endevor Quick Edit.

Note: The user can change defaults in effect for their own user ID on the Actions Options panel.

COMPONENTS_ONLY

Specifies whether the Delete action deletes the element component list.

Y – Delete processing deletes the element component list.

N – Delete processing deletes the element component list and the element.

Note: This option also affects standard CA Endevor SCM delete processing.

DELETE_MODE

Specifies whether the Delete action is executed in foreground or batch.

F – Delete processing is performed in foreground. This is the default.

B – Delete processing is performed in batch.

Note: This option also affects standard CA Endevor SCM delete processing.

c. The following parameters affect Move action processing in CA Endevor Quick Edit.

Note: The user can change defaults in effect for their own user ID on the Actions Options panel.

SYNCHRONIZE

Specifies whether to create a synchronization level as part of move or transfer processing.

Y – Move processing creates a synchronization level when an element is moved to a location where an out-of-sync condition exists.

N – Move processing does not create a synchronization level when an element is moved.

Note: This option also affects standard CA Endevor SCM processing.

WITH_HISTORY

Specifies whether an element is moved with history.

Y – Moves the element with history.

N – Moves the element without history. This is the default.

Note: This option also affects standard CA Endevor SCM processing.

RETAIN_SIGNOUT

Specifies whether CA Endevor SCM retains the signout associated with an element at the source location when it is moved to the target location.

Y – Retains the source location signout at the target location.

N – Does not retain the source location signout at the target location.

Note: This option also affects standard CA Endevor SCM processing.

ACKNOWLEDGE_ELEMENT_JUMP

Specifies whether a message is issued in a jump situation. A jump situation occurs when a copy of an element being moved is found at a non-mapped stage between the FROM and TO locations of the Move. In a jump situation, CA Endevor Quick Edit examines the REQ ELM JUMP ACKNOWLEDGEMENT field in the System definition.

Y – Issues a message when an element is jumped during a move, informing you of this fact. If the System definition setting for REQ ELM JUMP ACKNOWLEDGEMENT is set to Y, then Acknowledge Elm Jump must also be set to Y for the move action to complete.

N – Does not issue a message when an element is jumped during a move. If the System definition setting for REQ ELM JUMP ACKNOWLEDGEMENT is set to Y, then the move is not completed. This is the default.

Note: This option also affects standard CA Endevor SCM processing.

DELETE_AFTER_MOVE

Specifies whether the element is deleted at the source location after it is moved.

Y – Deletes the element at the source location after it is moved.

N – Does not delete the element at the source location.

Note: This option also affects standard CA Endevor SCM processing.

MOVE_MODE

Specifies whether the Move action is executed in foreground or batch.

F – Move processing will be performed in foreground. This is the default.

B – Move processing will be performed in batch.

Note: This option also affects standard CA Endevor SCM processing.

d. The following parameters affect the selection lists returned.

Note: The user can change these values on the CA Endevor Quick-Edit panel.

QE_BUILD_USING_ MAP

Specifies which Environments and Stages are included in a selection list.

Y – The selection list includes elements from all Stages and Environment in the map in which the specified Environment resides. This is the default.

N – The selection list includes elements in Stage 1 and Stage 2 of the specified Environment.

QE_RETURN_ FIRST_FOUND

Specifies whether the display includes only the first occurrence of an element found in the map.

Y – Returns only the first occurrence of the requested element. This is the default.

N – Returns all occurrences of the requested elements found along the map route.

e. The following parameters affect sign in and sign out processing in CA Endevor Quick Edit.

Note: The user cannot change these defaults.

OVERRIDE_ SIGNOUT

Specifies whether the signout status of elements can be overridden.

Y – You can retrieve an element and have it signed out to yourself, even if the element is signed out to someone else.

N – An element cannot be signed out to you, if the element is already signed out to someone else. This is the default.

Note: This option also affects standard CA Endevor SCM processing.

SIGNIN_MODE

Specifies whether the signin action executes in foreground or batch.

F – Signin processing will be performed in foreground. This is the default.

B – Signin processing will be performed in batch.

Note: This option also affects standard CA Endevor SCM processing.

SIGNOUT_MODE

Specifies whether the signout action executes in foreground or batch.

F – Signout processing will be performed in foreground. This is the default.

B – Signout processing will be performed in batch.

Note: This option also affects standard CA Endevor SCM processing.

The ISPF Configuration table is updated with your preferences.

2. Reassemble and link-edit the table.

   Your preferences for the default field values are set.

Edit Session Startup Command

The CA Endevor Quick Edit Startup command—ENDIEIM1—replaces the ISPF Edit Service's Initial Macro command, which is not supported by CA Endevor Quick Edit. The command is executed after CA Endevor Quick Edit invokes the Edit Service, but before the Edit Service displays the ISPF Edit panel. All ISPF/PDF Edit Macro services are available.

You can use the CA Endevor Quick Edit Startup command to do the following:

- Define aliases for user-written Edit macros.
- Set the Edit profile based on the element Type. The sample startup command contains an example of how to set the Edit profile name.
- Write additional information messages.
- Invoke a non-standard editor, for example SDF2's editor, to modify an element, without invoking the standard editor.

How to Write the Edit Session Startup Command

You can write the Edit session startup command in REXX or CLIST. A sample startup command is provided with the installation files, in the *iprfx.iqual*.CSIQCLS0 data set.

ISPF Dialog Variables for Start Up Command

Before CA Endevor Quick Edit invokes the startup command, it populates a set of ISPF dialog variables in the profile pool. The startup command can use the ISPF VGET service to retrieve the variables from the profile pool and use them as necessary. Any changes that the startup command makes to the variables are ignored.

The following identifies the variables that are available:

ENVBENV

Specifies an up to eight-character environment.

ENVBSYS

Specifies an up to eight-character base system.

ENVBSBS

Specifies an up to eight-character base subsystem.

ENVBTYP

Specifies an up to eight-character base type.

ENVBSTGI

Specifies a one-character base stage ID.

ENVBSTGN

Specifies a base stage number. Value values are 1 or 2.

ENVSENV

Specifies an up to eight-character source environment.

ENVSSYS

Specifies an up to eight-character source system.

ENVSSBS

Specifies an up to eight-character source subsystem.

ENVSTYP

Specifies an up to eight-character source type

ENVSSTGI

Specifies a one-character source stage ID.

ENVSSTGN

Specifies a source stage number. Valid values are 1 or 2.

ENVELM

Specifies an up to ten-character element name.

ENVPRGRP

Specifies an up to eight-character processor group.

ENVCCID

Specifies an up to twelve-character CCID.

ENVCOM

Specifies an up to forty-character comment.

ENVOSIGN

Specifies a signout value. Valid values are Y or N.

ENVGENE

Specifies a generate element value. Valid values are Y or N.

ENDIEIM1 Return Code Settings

The CA Endevor SCM Edit session start-up command, ENDIEIM1, can pass back a special return code to CA Endevor Quick Edit that has the follows effects:

- If ENDIEIM1 sets the return code to 20, CA Endevor Quick-Edit does *not* enter an Edit session and the modified source is saved in the entry stage.
- If ENDIEIM1 sets the return code to 24, CA Endevor Quick-Edit does *not* enter an Edit session and the current action is cancelled.
- For all other return codes, CA Endevor Quick-Edit opens an Edit session showing the current copy of the element.

Review ENDES000

The member ENDES000 in CSIQSENU contains the skeleton SCL that is used by CA Endevor Quick Edit when you execute the Generate Processor in batch.

Note: For more information about generating in batch, see Generate Processing.

Review ENDES000 and tailor it to meet the requirements of your site as needed.

Note: You may not need to change anything, as an earlier installation step may have made these updates to this member.

CONLIB DD DSN=iprfx.iqual.CSIQLOAD,DISP=SHR

iprfx.iqual

C1TPDD01 DD UNIT=tdisk

tdisk

C1TPDD02 DD UNIT=tdisk

tdisk

C1TPLSIN DD UNIT=tdisk

tdisk

C1TPLSOU DD UNIT=tdisk

tdisk

CA Endevor SCM Exit Support

CA Endevor Quick Edit supports the following CA Endevor SCM exit points:

- Exit 1—Security
- Exit 2—Before Action
- Exit 3—After Action
- Exit 5—Initialization
- Exit 6—Termination

Exits 5 and 6 occur automatically—Exit 5 before CA Endevor Quick Edit is invoked and Exit 6 after CA Endevor Quick Edit is invoked.

Note: For more information about each exit, see the *Exits Guide*.

CA Endevor Quick Edit handles Exits 2 and 3 as follows:

Edit (Update)

Note: If the element exists in the target Stage, Quick Edit will set UPDATE. However, if the element does not exist in the target Stage, then Quick Edit will set ADD.

- Exit 2 is invoked before element fetch
- Exit 3 is invoked after the action, regardless of how Exit 2 completed

  Note: If you use the Save command multiple times during an edit session, all the saves together are considered to be one action. The Save command does not invoke Exit 3. Exit 3 is called only when you use the End or CANCEL command to complete the action.

  Note: The ENCOPTBL option EXIT2_FAIL_EXIT3 allows you to specify that, in the event that Exit 2 fails, Exit 3 not be invoked. For more information, see the *Administration Guide*.

Create (Add)

- Exit 2 is invoked before invoking the Edit Service
- Exit 3 is invoked after the action, regardless of how Exit 2 completed

  Note: The ENCOPTBL option EXIT2_FAIL_EXIT3 allows you to specify that, in the event that Exit 2 fails, Exit 3 not be invoked. For more information, see the *Administration Guide*.

Generate (Generate)

- Exit 2 is invoked before invoking the Generate Processor
- Exit 3 is invoked after the action, regardless of how Exit 2 completed

  Note: The ENCOPTBL option EXIT2_FAIL_EXIT3 allows you to specify that, in the event that Exit 2 fails, Exit 3 not be invoked. For more information, see the *Administration Guide*.

Appendix D: Web Services

Note: The content of the appendix "Web Services" has been moved to the scenario-based knowledge document *How to Enable Web Services*, which can be found in the *Scenario Guide*.

Appendix E: CA Endevor Quick Edit Eclipse-Based UI

This section contains software requirements for the Eclipse-Based UI and instructions on how to install the Eclipse-Based UI on client computers.

Before you can install the Eclipse-Based UI, you must install and configure Web Services. For more information, see CA Endevor SCM Web Services (see page 193).

The Web Services installation process automatically installs the Eclipse-Based UI update site on your Tomcat server in the folder /Eclipse-BasedUI. You will access this update site, when you install the Eclipse-Based UI on your client computer.

This section contains the following topics:

Software Requirements for the Eclipse-Based UI

The following software is required for the Eclipse-Based UI:

- CA Endevor SCM Version 17.0.00.
- CA Endevor Quick Edit Version 17.0.00.
- CA Endevor SCM Version 17.0.00 component Web Services. For information on how to install and configure the Web Services component, see the scenario-based knowledge document *How to Enable Web Services,* which can be found in the *Scenario Guide.*
- The Eclipse Foundation's Eclipse SDK must be installed on each client computer where the Eclipse-Based UI is to be installed. The Eclipse-Based UI has been certified with release 4.3 of the Eclipse Foundation's Eclipse SDK.

  Note: The Eclipse-Based UI can be installed on platforms that support the Eclipse SDK version you are using.

  The Eclipse SDK includes the Eclipse Platform, Java development tools, and Plug-in Development Environment, which contains source code, user documentation, and development documentation. Eclipse does not include a Java Runtime Environment (JRE). For the proper version of the JRE or Java Development Kit (JDK) that must be installed on your machine to run Eclipse, go to: http://wiki.eclipse.org/index.php/Eclipse_FAQs.

Install the Eclipse-Based UI on a Client Computer

The Web Services installation process automatically installs the Eclipse-Based UI update site on your Tomcat server in the folder /Eclipse-BasedUI. This procedure installs the Eclipse-Based UI on your client computer from the update site.

Note: If you prefer *not* to install directly from your Tomcat server, you can download a compressed update site from *<protocol>//<servername>:<portnumber>*/Eclipse-BasedUI/bin. This bin file contains the Eclipse-Based UI update site. After downloading the file, you can use the Eclipse Update Manager to define the local/archive update site repository.

The Eclipse-Based UI is installed on client computers using the Eclipse Update Manager, a feature provided by Eclipse. The Eclipse Update Manager is a built-in plug-in installation and update manager that simplifies the installation process and lets you manage various plug-in instances.

Note: For details on managing your configuration, see the Eclipse help.

To install the Eclipse-Based UI for use with Eclipse release 4.3

1. In Eclipse, select Help, Install New Software.

   The installation wizard opens.

2. Click Add.

   The Add Site dialog opens.

3. Complete the following fields and then click OK.

   - In the Name field, specify the name of the update site.
   - In the Location field, enter one of the following URLs and then click OK. Enter the URL depending on which protocol your web server is using. This default URL points to the CA update site for the CA Endevor SCM Eclipse-Based UI.
     - https://*<servername>:<portnumber>*/Eclipse-BasedUI
     - http://*<servername>:<portnumber>*/Eclipse-BasedUI

   The new CA Endevor SCM Eclipse-Based UI update site is added to the list of available software sites, and is automatically scanned for possible updates, which are listed in the Installation wizard.

4. Select the check box next to Eclipse-Based UI, and click Next to proceed with installation of the Eclipse-Based UI.

   a. When prompted to confirm the installation, click Finish.

   b. When prompted to restart Eclipse, click Yes.

   The Eclipse-Based UI is installed on the client computer. Eclipse restarts and the Eclipse-Based UI is accessible from Eclipse.

Install the Eclipse-Based UI from Archive File

Instead of updating directly from the update site on the Tomcat server, you can download a compressed version of this update site. Then, users can access this file to install the Eclipse-Based UI. To use the archive file, do the following.

1. Download the com.ca.endevor_[*build number*].bin.dist.zip.

   Right-click and select Save As: com.ca.endevor_[*build number*].bin.dist.zip

   Store this file in a local or networked location where users can access it for installation. End-users can install the Eclipse-Based UI from the local archive file from the Eclipse environment.

2. To install the Eclipse-based UI, complete the steps in the section Install the Eclipse-Based UI on a Client Computer (see page 196) with the following exception:

   In step 3, instead of setting the update site URL, click the Local or Archive button and navigate to location of the downloaded archived update site.

   - Use the Archive button, if you are selecting the zip file.
   - Use the Local button, if the archived update site was decompressed.

   Then, follow rest of the instructions in the section Install the Eclipse-Based UI on a Client Computer.

Edit Proxy Definitions for Network Connectivity

To enable Eclipse IDE to connect to the Internet and pass through a proxy server, you need to edit the Eclipse proxy definitions. The Active Provider settings specify the profile to be used when opening connections and has the following options:

- The Direct option for Active Provider causes all the connections to be opened without the use of a proxy server.
- On some platforms, there is also a Native option that causes settings that were discovered in the OS to be used.
- The Manual option for Active Provider lets you specify proxy settings and causes those settings to be used.

This procedure uses the Manual option.

To edit proxy definitions

1. Open the Window menu on the Eclipse main menu and then select Preferences.

   The Preferences window opens.

2. Expand the General tree and select Network Connections.

   The Network Connections window opens.

3. Select Manual in the Active Provider drop down list. Then click on the row with the protocol you want to edit, for example HTTP. Then click the Edit button.

   The Edit Proxy Entry dialog opens for the protocol you selected.

4. Complete the Host and Port fields with information relevant for your site. If Authentication is required, check the Requires Authentication box and fill in the User and Password fields. Then click Ok.

   The Edit Proxy Entry dialog closes and the Network Connections window opens.

5. Click Apply and then OK.

   The proxy settings are saved and the Network Connections window closes.

6. Restart Eclipse.

   Any changes that you saved to the proxy settings are in effect.

Note: For more information, see the Eclipse IDE online help.

Appendix F: CA Endevor Integration for the Natural Environment

This section contains the following topics:

CA Endevor Integration for the Natural Environment

CA Endevor SCM Integration for the Natural Environment lets the Natural developer perform most, if not all, of their CA Endevor SCM tasks from the Natural development environment. CA Endevor SCM is used as a back-end repository for Natural while allowing the developer to control updates, migrations, and even the creation of change packages from the Natural environment.

The integration software enables application developers working in the Natural environment to perform CA Endevor SCM actions on Natural and Predict objects (known as elements).

This option runs under z/OS, within the CICS or TSO/ISPF environments and in batch. It accesses CA Endevor SCM through the CA Endevor SCM API.

Introduction

To deploy CA Endevor SCM Integration for the Natural Environment, the delivered data sets must be installed in the Natural and Adabas environments and Natural and CA Endevor SCM must be configured. These tasks are usually performed by the following personnel:

- The Natural administrator installs the Natural objects in the Natural environment.
- The Adabas administrator installs the Adabas file in Adabas.
- The CA Endevor SCM administrator configures the CA Endevor SCM environment.
- The Natural administrator and the CA Endevor SCM administrator work together to configure the Natural software lifecycle and set up the software inventory requirements.

Note: For more information about the files delivered with the product, see Components (see page 201).

Hardware Requirements

The following environments are supported:

- IBM-supported releases of z/OS through V1.13

Software Requirements

The following mainframe software is required:

- Adabas v7.4 with Natural v3.1.6 or v4.1.
- Adabas v8.1 or v8.2 with Natural v4.1 or v4.2 or v8.2
- (Optional) Predict v4.x
- CA Endevor SCM Version 15.0, 15.1, or 16.

Concurrent Releases

Concurrent releases of CA Endevor SCM Integration for the Natural Environment can be run when installed into different Natural libraries. Each instance of the application should use their own Adabas files.

Components

The following components are delivered with CA Endevor SCM Integration for the Natural Environment:

- CA Endevor SCM Integration for the Natural Environment executable files

  The following executables are supplied:

  - The integration software's online interface in the Natural environment.
  - The integration software's server, which usually runs as a z/OS started task, serving as an interface between the integration software's online sessions and the CA Endevor SCM API.

- Natural

  The Natural EINE program initiates CA Endevor SCM Integration for the Natural Environment sessions. The Natural EINE module should be placed in the SYSTEM library to allow users to initiate sessions without needing to first log on to the integration software's main library. The source for this program is delivered to allow you to run concurrent versions of the integration software.

- Adabas files

  The following Adabas files are supplied:

  - message queue file

    This file serves as the message queue between the integration software's online sessions (which write requests to and read responses from the message queue file) and integration software's servers (which read requests from and write responses to the message queue file). The message queue file also provides the means for an administrator to issue a shutdown command to a CA Endevor SCM Integration for the Natural Environment server.

  - system profile file

    This file serves as a repository for all non-message system data, including user profile data.

- CA Endevor SCM processor templates

  The following processor templates are supplied:

  - One set (generate, move, and delete) providing for Natural objects
  - One set (generate, move, and delete) providing for Predict objects

- SCL

  The supplied SCL must be customized and run to define Natural and Predict element types to CA Endevor SCM, and to define processor groups and symbolic variables to those element types.

- Global Type Sequencing Type Statements

  Type statements are supplied for inclusion into your site's global type sequencing scheme, providing for Predict files, Natural DDMs, and Natural programming objects to be processed in the correct order by CA Endevor SCM packages and batch processes.

- $ESYMBOL samples and template

  The $ESYMBOL samples and templates are used to configure the integration software's Site Symbolics, which are used in Natural and Predict processors, and which are also used by the integration software in the generation of JCL (for batch submission of CA Endevor SCM requests).

- Sample JCL

  The sample JCL is used to install the integration software and configure and run instances of the integration software's server (as started tasks).

How to Install the Software

To enable CA Endevor SCM to control application development in the Natural environment, CA Endevor SCM Integration for the Natural Environment needs to be installed in your Natural and Adabas environments and CA Endevor SCM must be appropriately configured. The installation and configuration process includes the following steps.

1. Install the CA Endevor SCM Integration for the Natural Environment data sets on the mainframe.
2. Install the Natural objects into the Natural environment. (see page 203) (Done by the Natural administrator).
3. Install the Adabas files into the Adabas environment. (see page 204) (Done by the Adabas administrator).
4. Edit, assemble, and relink the NATPARM module for the Natural nucleus. (see page 205) (Done by the Natural administrator).
5. Install user exits (see page 209). (Done by the Natural administrator.)
6. Create a default job card for batch jobs. (see page 218) (Done by the CA Endevor SCM administrator.)
7. Create a default CA Endevor SCM procedure for package execution. (see page 218) (Done by the CA Endevor SCM administrator).
8. Create a CA Endevor SCM procedure for action processing. (see page 219) (Done by the CA Endevor SCMadministrator).
9. Configure a Natural Lifecycle in CA Endevor SCM (see page 220). (Done by the CA Endevor SCM administrator).

10. Configure the CA Endevor Integration for the Natural Environment server startup (see page 237) and shutdown (see page 238) tasks. (Done by the Natural administrator.)
11. Update C1DEFLTS— Make sure that the CA Endevor Automated Configuration ACM control option, ASCM= is set to Y in the Defaults table (C1DEFLTS). For more information about updating the C1DEFLTS, see the chapter "Using the Defaults Table" in the *Administration Guide*. (Done by the CA Endevor SCM administrator.)

Install the Natural Objects

You must install the Natural objects and SYSERR messages in the Natural environment to enable CA Endevor SCM Integration for the Natural Environment. The Natural administrator usually performs this procedure; however, if Natural Security is installed at your site, the Natural Security administrator performs step 1.

Important: If you are upgrading from a previous version of CA Endevor Integration for the Natural Environment, copy the NDVJOBCD, NDVPROCP, UEXIT00, UEXIT99 (if modified), and any NAT* TEXT members that have been modified to a temporary library. All members of the EINELIB will be replaced by LOADNAT job.

If you want to run the new version simultaneously with the current version, in the LOADNAT JCL use the NEWLIBRARY option to specify a different library name other than EINELIB, such as EINEVxxx.

To Install the Natural Objects

1. Define the EINE application to Natural Security if the Natural Security (NSC) is installed at your site; if Natural Security is not defined, skip this step and go to step 2. Define EINE with the Predict XREF feature set to OFF, because the EINE NATLOAD data set contains no XREF data. (This step is performed by the Natural Security administrator).

   This enables Natural Security to recognize the CA Endevor SCM Integration for the Natural Environment library and allows CA Endevor SCM Integration for the Natural Environment processing.

2. Execute the Natural LOADNAT JCL to load the CA Endevor SCM Integration for the Natural Environment programs to a FUSER file. The LOADNAT JCL is provided in the JCL library. This JCL does not run *as is* and is provided only as a reference.

   The LOADNAT JCL loads the CA Endevor SCM Integration for the Natural Environment programs and SYSERR messages into the EINELIB library.

   Note: You can also use the online Object Handler utility (SYSOBJH) to load the CA Endevor SCM Integration for the Natural Environment programs to a FUSER file. For more information about SYSOBJH, see the *Natural Utilities Guide*.

3. Copy the EINE program to the FUSER SYSTEM library to enable users to initiate sessions without logging on to the integration software's main library.

Install the Adabas Files

To enable CA Endevor Integration for the Natural Environment, you must install the message queue file in Adabas. The system profile file must be installed to store values set by the user from session to session, that is, user profile information.

Each instance of CA Endevor Integration for the Natural Environment must have its own unique set of files. The Adabas administrator usually installs these files using the following procedures.

Important! If you are upgrading from Version 12, 15.0, or 15.1, you must delete the EINE message queue file (file 125 only) and use the LOADADA JCL to allocate and load a new message queue file.

To Install the Adabas files

1. Edit the sample LOADADA JCL, provided in the JCL library, to meet your site standards. You can directly input the ADACMP data set provided in the SAGLOAD library into the Adabas LOADADA JCL.

   If you are upgrading from Version 12, 15.0, or 15.1, delete the LODSYS step from the LOADADA JCL. If you delete and reallocate the EINE system profile file (file 126), you will lose the user profile information stored in the system profile file.

   Note: The Natural transaction used to execute CA Endevor SCM Integration for the Natural Environment must be able to issue Adabas commands to the database where these files are installed.

2. (Optional) Edit the LOADADA JCL to use the Adabas Security facilities to password-protect and cipher-encrypt the files.

3. Execute the LOADADA JCL.

   The message queue file and the system profile file are added to the Adabas database where the CA Endevor SCM Integration for the Natural Environment modules are installed.

Edit, Assemble, and Relink the NATPARM Module

Edit the NATPARM module for each online and batch Natural nucleus used by CA Endevor SCM Integration for the Natural Environment. The Natural administrator usually performs this procedure.

To edit, assemble, and relink the NATPARM module

1. Insert a macro invocation after the last line in the NATPARM macro, but before the END statement, to support the message queue file. Code the macro invocation depending on which version of Natural you are using.

   NTFILE—Natural 3.1 Macro Invocation for NATPARM

   If you are using Natural 3.1, insert the following macro invocation in the NATPARM. This keyword has the following format:

   ```
   NTFILE ID=125,DBID=daa,FNR=faa,PASSW=password,CIPH=cipher-key
   ```

   ID=125

   Defines the file ID used internally by CA Endevor SCM Integration for the Natural Environment. It does not affect any file on the database that has a file number 125.

   DBID=*daa*

   Defines the Adabas DBID of the CA Endevor SCM Integration for the Natural Environment file.

   FNR=*faa*

   Defines the file number of the CA Endevor SCM Integration for the Natural Environment file.

   PASSW=*password*

   (Optional) Adabas security password.

   CIPH=*cipher-key*

   (Optional) Adabas security cipher-key.

NTLFILE—Natural 4.x Macro Invocation for NATPARM

If you are using Natural 4.x, insert the following macro invocation. This keyword has the following format:

```
NTLFILE 125,daa,faa,password,cipher-key
```

125

Defines the file ID used internally by CA Endevor SCM Integration for the Natural Environment. It does not affect any file on the database that has a file number 125.

daa

Defines the Adabas DBID of the CA Endevor SCM Integration for the Natural Environment file.

faa

Defines the file number of the CA Endevor SCM Integration for the Natural Environment file.

password

(Optional) Adabas security password.

cipher-key

(Optional) Adabas security cipher-key.

2. Insert a macro invocation after the last line in the NATPARM macro, but before the END statement, to support the system profile file. Code the macro invocation depending upon which version of Natural you are using.

   NTFILE_Natural 3.1 Macro Invocation for NATPARM

   If you are using Natural 3.1, insert the following macro invocation in the NATPARM. This keyword has the following format:

   ```
   NTFILE ID=126,DBID=daa,FNR=faa,PASSW=password,CIPH=cipher-key
   ```

   ID=126

   Defines the file ID used internally by CA Endevor SCM Integration for the Natural Environment. It does not affect any file on the database that has a file number 126.

   DBID=*daa*

   Defines the Adabas DBID of the CA Endevor SCM Integration for the Natural Environment file.

   FNR=*faa*

   Defines the file number of the CA Endevor SCM Integration for the Natural Environment file.

   PASSW=*password*

   (Optional) Adabas security password.

   CIPH=*cipher-key*

   (Optional) Adabas security cipher-key.

NTLFILE—Natural 4.x Macro Invocation for NATPARM

If you are using Natural 4.x, insert the following macro invocation. This keyword has the following format:

```
NTLFILE 126,daa,faa,password,cipher-key
```

126

Defines the file ID used internally by CA Endevor SCM Integration for the Natural Environment. It does not affect any file on the database that has a file number 126.

daa

Defines the Adabas DBID of the CA Endevor SCM Integration for the Natural Environment file.

faa

Defines the file number of the CA Endevor SCM Integration for the Natural Environment file.

password

(Optional) Adabas security password.

cipher-key

(Optional) Adabas security cipher-key.

3. If you are using local buffer pools, verify that the size of the Natural buffer pool used for the System Object Handler (SYSOBJH) Natural utility is set to 5120 or higher. The size of the buffer pool is set by the SIZE parameter of the TYPE=NAT NTBPI macro in the NATPARM to be used for the CA Endevor Integration for the Natural Environment server (for example, NTBPI TYPE=NAT,SEQ=0,SIZE=5120). Failure to have a large enough buffer pool can result in NAT0777 errors when attempting to perform Add or Retrieve actions.
4. Assemble the NATPARM module using the standard NATPARM assembly procedure, if it was modified.

   The assembly should receive a condition code of 0.
5. Relink the modified NATPARM modules. No modifications are required to the standard JCL used to link Natural, and no additional INCLUDE statements are needed. If necessary, recycle the CICS or COM-PLETE region to load the new Natural executable module as resident.

Install User Exits

Certain user exits supplied by Software AG are required. In addition, optional exits are provided that allow you to customize security and exit procedures.

The Natural administrator usually performs this procedure.

To install the user exits

1. Copy the object code of the following Natural objects from the SYSEXT library to either the SYSTEM library or the EINELIB library (in the Natural environment in which CA Endevor SCM Integration for the Natural Environment was loaded). The following Software AG supplied user exits are required and must reside in either the SYSTEM library or the EINELIB library:

   USR0050N USR0210N USR0320N USR0420N USR1033N

   USR1043N USR1051N USR1054N USR1055N USR1056N

   USR1057N USR2004N USR2014N USR2018N USR2019N

   USR2027N USR3013N

2. (Optional) Modify the initialization exit. The CA Endevor Integration for the Natural Environment initialization exit (UEXIT00) is used to control configuration options for the EINE Natural application. The initialization exit is invoked (by fetch) immediately subsequent to invocation of a CA Endevor Integration for the Natural Environment session from the Natural command prompt.

   Modify UEXIT00 by logging on to the EINELIB Natural library, update UEXIT00 to your site requirements, and issue a STOW command. Once the modifications are complete and the exit STOWed, the EINE Servers must be shutdown and restarted.

   Note: If you are upgrading from Version 15.0 or Release 15.1 and saved your current UEXIT00 to a temporary Natural library, you can use it as a model for updating the Version 16.0 UEXIT00.

   You can make the following changes to UEXIT00:

   a. Add security checks to allow the session to be terminated from within the user-exit module.

   b. Change the values for the following variables if you want to change the default values:

      +UEXIT00-RJE-GEN-USERID-PASSWORD

      Defines the logical variable that is used to tell EINE whether or not to include USERID= and PASSWORD= as part of the job card for jobs submitted using RJE.

      Valid values are TRUE or FALSE. The default is TRUE.

+EINE-TIMEOUT-INTERVAL-SERVER

Defines the wait time limit for the server to begin processing a request. This is the time in seconds that the server is allowed to begin processing a queued request before returning control to the user. The user may choose to wait for the request or return to EINE. Timeouts for this value can indicate that additional servers may be required.

Valid values are 10 to 180 seconds. If the value is set outside this range, the default is used. The default is 30 seconds.

+EINE-TIMEOUT-INTERVAL-API

Defines the wait time limit that the API has to process a request from the server. This is the time in seconds that the API is given to process a request before returning control to the user. The user may choose to wait for the request to complete or return to EINE.

Valid values are 10 to 180 seconds. If the value is set outside this range, the default is used. The default is 30 seconds.

+EINE-TIMEOUT-INTERVAL-PWD

Defines the timeout interval for user passwords. A zero value causes a prompt for a password to occur for every logon attempt. If the value is greater than zero, the user is not prompted to reenter a password for each new session until the timeout interval has expired. The encrypted password is stored in the User Profile record and is used with the user ID to verify access to EINE. If the logon attempt fails, the logon window reopens.

Valid values are 0 to 999999. The value is defined in minutes (1440 = one day, 10080 = one week, 525600 = one year). The default is zero (0), which forces the user to log on every time upon entering EINE.

+EINE-CLEANUP-LATENCY-MINUTES

Defines the time in minutes after which a request found on the message queue file is deleted.

Any positive value is valid. The default is 240 minutes (four hours).

+EINE-MAX-ADD-RETRIEVE-PER-JOB

Defines the maximum number of Add or Retrieve actions to be processed by one batch job submitted by the EINE application. If more than this number is requested, multiple jobs are submitted.

Valid values are 1 to 240. The default is 240.

+EINE-STARTUP-CMD

Defines the name of the EINE program used to startup the EINE application.

The default is EINE.

+EINE-INSTALL-LIBRARY

Defines the name of the Natural library into which the EINE executables have been installed.

The default is EINELIB.

+EINE-SCL-LIBRARY

Defines the name of the Natural library into which an SCL Text Object is saved.

The default is EINESCL.

+EINE-SYSTRANS-SYSOBJH

Defines the Natural utility program used to load and unload Natural objects from Natural libraries.

Valid values are:

O— Specifies SYSOBJH Internal format which includes the Natural line numbers when the Natural object is unloaded and stored in CA Endevor SCM meta data files. This is the default.

F— Specifies SYSOBJH Transfer format which does not include the Natural line numbers when the Natural object is unloaded and stored in CA Endevor SCM meta data files.

L— Specifies SYSOBJH Transfer USE-LINE-NUMBER-INCREMENT which works the same as the Transfer option, F, except that the USE-LINE-NUMBER-INCREMENT keyword is specified in the Transfer clause. The USE-LINE-NUMBER-INCREMENT keyword causes the line number increment to be unloaded with the Natural object during the SYSOBJH unload process. If the line number increment is unloaded with the Natural object, it is used to rebuild the line numbers of the Natural source object during the SYSOBJH load process.

Note: The USE-LINE-NUMBER-INCREMENT option is only available with Natural 8.2.3 SP07 or higher.

T— Specifies SYSTRANS. The SYSTRANS utility is no longer supported in Natural v8.2 and higher. If currently using SYSTRANS, switch to the SYSOBJH Transfer format.

Note: For more information about how the +EINE-SYSTRANS-SYSOBJH setting affects line numbering, see the example at the end of this topic.

+EINE-RESTRICT-JOBCARDS

Determines if the Job Cards presented in the Batch Submittal window can be modified or not.

Valid values are 'Y' and 'N'. The default is N.

+EINE-LOGON-EXIT-ENABLED

Used to determine whether the EINE Logon exit, UEXIT01, is enabled or not. If +EINE-LOGON-EXIT-ENABLED is enabled (set to Y), then UEXIT01 is FETCHed during the EINE logon process. Sites can use UEXIT01 to perform any custom logon processing necessary.

Valid values are 'Y' and 'N'. The default is N.

+EINE-DSPLY-MENU-PACKAGES

Determines if the Packages Option is displayed on the EINE Main menu.

Valid values are Y or N. The default is Y.

+EINE-DSPLY-MENU-SAVEDSCL

Determines if the Saved SCL Option is displayed on the EINE Main menu.

Valid values are Y or N. The default is Y.

+EINE-DSPLY-MENU-DDM

Determines if the DDM Option is displayed on the Foreground and Batch Processing Menus.

Valid values are Y or N. The default is Y.

+EINE-DSPLY-MENU-PREDICT

Determines if the Predict Option is displayed on the Foreground and Batch Processing Menus.

Valid values are Y or N. The default is Y.

+EINE-DFLT-SIGNOUT

Defines the initial default Signout Element action option value for Retrieve actions.

Valid values are Y or N. The default is N.

+EINE-DFLT-REPLACE

Defines the initial default Replace if Present action option value for Retrieve actions.

Valid values are Y or N. The default is Y.

+EINE-DFLT-GENERATE

Defines the initial default Generate Element action option value for Add and Transfer actions.

Valid values are Y or N. The default is Y.

+EINE-DFLT-UPDATE

Defines the initial default Update if Present action option value for Add actions.

Valid values are Y or N. The default is Y.

+EINE-DFLT-W-HIST

Defines the initial default With History action option value for Move and Transfer actions.

Valid values are Y or N. The default is N.

+EINE-DFLT-RET-SIGNOUT

Defines the initial default Retain Signout action option value for Move and Transfer actions.

Valid values are Y or N. The default is N.

+EINE-DFLT-ACK-JUMP

Defines the initial default Acknowledge Jump action option value for Move actions.

Valid values are Y or N. The default is N.

+EINE-DFLT-DEL-FROM

Defines the initial default Delete from Element action option value for Move and Transfer actions.

Valid values are Y or N. The default is Y.

+EINE-DFLT-COPYBACK

Defines the initial default Copyback action option value for Generate actions.

Valid values are Y or N. The default is N.

Note: The +EINE-DFLT-*xxxxxx* parameters are used to establish site specific initial default values presented on the Foreground and Batch Processing screens for the CA Endevor SCM Action Options: Signout Element, Replace if Present, Generate Element, Update if Present, With History, Retain Signout, Acknowledge Jump, Delete 'From' Element, and Copyback. If the user modifies the value of one of these Action Options on an EINE Processing screen, the user specified value is retained in the user's System Profile record. That retained value is used the next time the Action Option is presented.

+EINE-DFLT-EW-FROM-DATE

Defines the default value to be used for the package Execution Window From date field when a package is created.

The default value of blank will cause the panel to display the current date in the Execution Window From date field.

+EINE-DFLT-EW-FROM-TIME

Defines the default value to be used for the package Execution Window From time field when a package is created.

The default value of blank will cause the panel to display 00:00 in the Execution Window From time field.

+EINE-DFLT-EW-TO-DATE

Defines the default value to be used for the package Execution Window To date field when a package is created.

The default value of blank will cause the panel to display 31DEC79 in the Execution Window To date field.

+EINE-DFLT-EW-TO-TIME

Defines the default value to be used for the package Execution Window To time field when a package is created.

The default value of blank will cause the panel to display 00:00 in the Execution Window To time field.

Note: The +EINE-DFLT-EW-*xxxx-xxxx* parameters are used to display default values for the package Execution Window From and To dates and times when a package is being created. These fields are not edited for content. If non-numeric values are coded for the Execution Window parameters, the user will need to enter valid execution window dates and times when creating a package. Usually, there is no need to provide a value other than blank for these parameters.

3. (Optional) Enable the CA Endevor Integration for the Natural Environment logon exit, UEXIT01, to customize any site specific logon procedures. UEXIT01 is called during the online session initialization and logon process. The exit is invoked after the user has logged on and their credentials have been authenticated, UEXIT00 has been initialized and validated, and processing of the Site Symbolics has completed. Use UEXIT01 to do the following:

   - Perform customized logon validation
   - Customize the job cards that are presented in the Batch Job
   - Customize the job cards that are presented in the Batch Job Submittal window specifically for the user logging on

     Note: The UEXIT00 +EINE-RESTRICT-JOBCARDS parameter can be used with this option to prevent users from modifying the job cards presented in the Batch Job Submittal window.
   - Abort a logon attempt

To enable UEXIT01, the UEXIT00 +EINE-LOGON-EXIT-ENABLED parameter must be set to Y. You must provide and write the logon processing logic for UEXIT01 and STOW the UEXIT01 exit in the EINE install library. The exit contains the following fields. Edit the fields as appropriate for your site.

- Input fields

  +UEXIT01-ENDEVOR-USER

  Contains a copy of the CA Endevor SCM user ID. The logged on CA Endevor SCM user ID cannot be changed. If +UEXIT01-ENDEVOR-USER is changed in the exit, the modification is ignored.

- Input/Output fields

  +UEXIT01-BATCH-JOBCARD

  Contains a copy of the four job card records read from the EINE System Profile file for the loggedon user. These job card records can be modified by the exit and if the return code is set to a 4, the user's System Profile record is updated with them and they are presented to the user in the Batch Job Submittal window.

- Output fields

  +UEXIT01-RETURN-CODE

  Specifies the action to be taken by EINE upon returning to the main login process.

  +UEXIT01-MESSAGE

  Specifies up to four text message lines that display on the UEXIT01 Abort Logon screen when the Return Code is set to 12.

  UEXIT01 Return Codes

  Specifies the action that the exit will perform depending on the return code received. Valid return codes for this exit are listed next. Place the appropriate code in the +UEXIT01-RETURN-CODE Independent variable field before returning.

  0— No updates made; proceed with logon.

  4— Modifications made, update the job card records in the user's System Profile record, and proceed with logon.

  12— Error; abort user logon.

  Any other value is treated as a Return Code 0

4. (Optional) Modify the CA Endevor Integration for the Natural Environment termination exit (UEXIT99) to add alternate or additional site-specific exit procedures. The termination exit, UEXIT99, is invoked (by fetch) when a user exits a session. By default, this user-exit program stacks a return command to provide for the user to be returned to the library from which the session originated.

   Modify UEXIT99 by logging on to the EINELIB Natural library, update UEXIT99 to your site requirements, and issue a STOW command. Once the modifications are complete and the exit STOWed, the EINE Servers must be shutdown and restarted.

   Note: If you are upgrading from Version 15.0 or Release 15.1 and saved your current UEXIT99 to a temporary Natural library, you can use it as a model for updating the Version 16.0 UEXIT99.

Example: Initialization Exit UEXIT00 Variable EINE SYSTRAN SYSOBJH Settings

When the initialization exit, UEXIT00, variable +EINE-SYSTRAN-SYSOBJH is set to **O**, Natural objects are unloaded in the SYSOBJH Internal format. The internal format includes the line numbers in hexadecimal with the unloaded objects. This causes CA Endevor SCM to identify lines as changed when only the line number changed as a result of lines being inserted or delete and the line numbers in the Natural object are automatically resequenced by the Natural editor. In this example, only the line in bold font was actually changed, but all lines thereafter are marked with a percent sign (%) to indicate that these lines changed also.

```
+0100      ??*  MAINTENANCE HISTORY:
+0100      ? *
+0100      ?&*  DATE    WHO WHAT
+0100      ?-*  ------- --- ----------------------------------------
+0100      ?ø*  15MAR88 RSC
+0100      ?Ø*  02MAY89 RSC CHANGED P6.5 FMTS TO P8.3
+0100      ?°*  16JUN98 JCH MODIFY FOR 4 DIGIT YEAR
+0100      ? *  14MAR01 JCH INCREASE NAT-PGMID TO 22 BYTES
%+0101     ??*  12AUG13 JAA INSERTING TEST COMMENT LINE ONLY
%+0101     ??************************************************************
%+0101     ??*
%+0101     ? DEFINE DATA
%+0101     ?&  GLOBAL USING HIST-GDA
%+0101     ?-END-DEFINE
%+0101     ?ø*
```

When the UEXIT00 variable +EINE-SYSTRAN-SYSOBJH is set to **F**, Natural objects are unloaded with the SYSOBJH Transfer option. The transfer option does not include line numbers with the unloaded objects. This results in only the lines of Natural code that actually changed being identified as changed by CA Endevor SCM. In this example, only the line in bold font was actually changed and it is the only line marked with a percent sign (%) to indicate the line changed.

```
  +0100      *S***  MAINTENANCE HISTORY:
  +0100      *S***
  +0100      *S***  DATE    WHO WHAT
  +0100      *S***  ------- --- ------------------------------------
  +0100      *S***  15MAR88 RSC
  +0100      *S***  02MAY89 RSC CHANGED P6.5 FMTS TO P8.3
  +0100      *S***  18JUN98 JCH MODIFY FOR 4 DIGIT YEAR
  +0100      *S***  14MAR01 JCH INCREASE NAT-PGMID TO 22 BYTES
 %+0101      *S***  14AUG13 JAA ADDED THIS COMMENT LINE ONLY
  +0100      *S*****************************************************
  +0100      *S***
  +0100      *S***  THIS VERSION READS ALL NATL RECORDS WITHIN A TIME
  +0100      *S***  REJECTS THOSE NOT NEEDED, DOES A -SORT- TO GET THE
  +0100      *S***  RECORDS INTO PGM SEQUENCE, THEN SUMMARIZES THE REC
  +0100      *S***  EACH NATURAL PROGRAM.
  +0100      *S***
```

When the UEXIT00 variable +EINE-SYSTRAN-SYSOBJH is set to **L**, Natural objects are unloaded with the SYSOBJH Transfer option with the USE-LINE-NUMBER-INCREMENT keyword. The TRANSFER USE-LINE-NUMBER-INCREMENT (L) option has the same behavior in CA Endevor SCM as the Transfer option (F) where the line numbers are not included in the unloaded source, so that only the lines of Natural code that actually changed are identified as changed by CA Endevor SCM. The difference is that the TRANSFER USE-LINE-NUMBER-INCREMENT (L) option includes the line number increment with the object when it is unloaded and then the line number increment is used to rebuild the line numbers of the Natural object during the SYSOBJH load process.

Create a Default Job Card for Batch Requests

You can create a default job card for initial display in the Batch Job Submittal window. After a user modifies the job card information, it is retained in the system profile file for that user.

Note: If you are upgrading from Version 15.0 or Release 15.1 and saved your NDVJOBCD to a temporary Natural library, you can use it as a model for updating the Version 16.0 NDVJOBCD.

To create a default job card for batch requests

1. Log on to the EINELIB Natural library.
2. Modify the text module NDVJOBCD to make it consistent with your general site standards for a job card layout. Include up to four lines as the default job card used when submitting batch jobs from CA Endevor SCM Integration for the Natural Environment.

   Note: NDVJOBCD contains the values #USERID, #USER7, or both. #USERID is automatically replaced with the submitting user's user ID, which is the Natural system variable *USER. #USERID7 is automatically replaced with the first 7 characters of the submitting user's user ID. Within the default job-card layout, these substitution values #USERID and #USER7 may optionally be followed by a period.

   The default job card is read from the first three lines of the text module NDVJOBCD.
3. Save the text module. When you submit batch requests, the text module NDVJOBCD is used as the default job card.

Create a Default CA Endevor SCM Procedure for Package Execution

This CA Endevor SCM procedure is used to build the JCL required for package execution.

Note: If you are upgrading from Version 15.0 or Release 15.1 and saved your NDVPROCP to a temporary Natural library, you can use it as a model for updating the Version 16.0 NDVPROCP.

To create a default CA Endevor SCM procedure for package execution

1. Log on to the EINELIB Natural library.
2. Modify the text module NDVPROCP to make it consistent with your general site standards for a package execution and save the text module.

   The text module NDVPROCP is available to build default JCL for package execution.

Create a CA Endevor SCM Procedure for Action Processing

A CA Endevor SCM procedure is needed for execution of CA Endevor SCM action processing. If your site has not already customized this procedure, use the following steps to create the procedure.

1. Customize the ENDEVOR procedure. A sample ENDEVOR procedure is located in the CSIQJCL library delivered with the CA Endevor SCM files.
2. After modification, copy it to a PROCLIB defined to JES.
3. If you changed the ENDEVOR member name, you must update the text module NDVSTEP in the EINELIB Natural library. To do this, perform the following steps:
   a. Log on to the EINELIB Natural library.
   b. Modify the text module NDVSTEP to update the PROCNAME value in the EXEC statement to reflect the new procedure name.
   c. Save the text module.

The Natural Development Lifecycle

The Natural administrator and the CA Endevor SCM administrator must work together to design and implement a Natural lifecycle. This task includes implementing a Natural development lifecycle, an element classification scheme, and defining the physical data sets in CA Endevor SCM to enable CA Endevor SCM to manage the Natural development process. CA Endevor SCM identifies Natural objects (also known as elements) according to the logical structure implemented at your site. All elements are identified by a fully-qualified name consisting of environment, stage, system, subsystem, type, and element name. Each element is defined by the following:

- Location in the software lifecycle. This is determined by the environment and stage where the element resides. As an application is modified, elements are moved through the software development lifecycle, with those elements residing in different functional locations at different times. Those functional areas, for example, Test, QA, Production, Backup, and so on, are defined as an environment and stage combination. The location of an element is part of the identification of that element.
- Inventory classification. This is determined by the system, subsystem, and type with which the element is associated.

A physical data set structure must be set up at your site to support your logical structure. For more information on defining your site's logical or physical structure, see the *CA Endevor Software Change Manager Administration Guide*.

Note: For more information about lifecycle concepts, see A Typical Software Lifecycle in the chapter "Understanding the Software" in the *User Guide*.

How to Configure a Natural Lifecycle in CA Endevor SCM

You must configure CA Endevor SCM to enable it to control the Natural development process. The Natural administrator and the CA Endevor SCM administrator need to work together to complete this task. To configure CA Endevor SCM to control Natural development, complete the following steps. These steps assume that CA Endevor SCM is already set up at your site. If this is not the case, see additional configuration requirements described in the *CA Endevor Software Change Manager Implementation Guide* and the *CA Endevor Software Change Manager Administration Guide*.

1. Identify the stages in the development lifecycle of your Natural code that you want to put under control of CA Endevor SCM.
2. Group the stages into environments, with two stages in each environment and one VSAM master control file for each stage.
3. Define the promotion route that inventory travels through the lifecycle, by mapping the stages in the C1DEFLTS table.
4. Create CA Endevor SCM libraries. See How to Create CA Endevor SCM Base, Delta, and List Libraries (see page 221).
5. Update the Optional Features table (ENCOPTBL) to activate the CA Endevor Integration for the Natural Environment option. For more information, see Edit, Reassemble, and Relink ENCOPTBL (see page 222).
6. Define site symbolics.
   a. Define site symbolics to associate each CA Endevor SCM stage to a specific FUSER/FDIC combination. You can use either the supplied sample JCL SITESYME, which provides a sample set of definitions, or SITESYMT, which is a template for definitions. For more information, see Associate Each CA Endevor SCM Stage to a Specific Natural FUSER/FDIC Combination (see page 223).
   b. Define site symbolics to associate your Natural libraries to CA Endevor SCM locations. For more information, see Methods of Associating Natural Libraries to CA Endevor SCM Inventory Locations (see page 226).
7. Set up the environments: assemble and link C1DEFLTS; assemble and link the Symbols table; and allocate the CA Endevor SCM VSAM master control files.
8. Define your systems. Identify functional groupings for your Natural applications, such as Finance applications, Tax applications, and so on. Define a system for each functional group and define each system to each environment in which you plan to use it. Define processor load and listing libraries for each system at both stages in an environment. These libraries can be shared across systems and environments.

9. Allocate the processor load and listing libraries, with one load and one listing library for each stage. For more information about allocating processor load and listing libraries, see the *CA Endevor Software Change Manager Administration Guide*.
10. Define your subsystems. Decide which Natural libraries and Predict files you want to manage in CA Endevor SCM. Define a subsystem for each Natural library, with at least one subsystem for each system. For example, your Finance system could have a subsystem for accounts payable (ACTTPAY) and one for accounts receivables (ACTTREC).
11. Create the CA Endevor SCM inventory and the Type PROCESS. Create a batch job that defines the systems, subsystems, and the Type PROCESS.
12. Edit the sample processors provided in the Processor Template data set. Add the processors to CA Endevor SCM using the sample JCL DEFPROC. For more information about installing processors, see Install the Processors (see page 230).
13. Edit your type and processor group definitions and create them in CA Endevor SCM using the sample JCL DEFTYPE. For more information, see Define the Natural and Predict Types (see page 233).
14. Enable Global Type Sequencing to define the sequence in which types are processed. You must use the supplied JCL member ETYPESEQ to order the Natural types. For more information, see Enable Global Type Processing.

How to Create CA Endevor SCM Base, Delta, and List Libraries

To implement an inventory structure for CA Endevor SCM Integration for the Natural Environment, you must define and allocate base, delta, and list libraries. The base, delta, and processor load and listing libraries are used during CA Endevor SCM execution. Base libraries store the original or current source version of an element. Delta libraries track the differences between the numerous source levels of an element. The processor load and listing libraries defined for a System hold the executable form of the processor (load) and a listing of the compile run for each processor. You can have multiple occurrences of the libraries, usually one set of libraries for each stage. For more information on base, delta and list libraries, see the *CA Endevor Software Change Manager Administration Guide*. The following steps are required to create the CA Endevor SCM base, delta, and list libraries.

1. Allocate data sets for the CA Endevor SCM base and delta libraries for your Natural element types. Define the PDS data sets with an RECFM value of VB and an LRECL value of at least 256. Make sure that you use your site naming conventions. All Natural element types can share a single pair of base and delta libraries.
2. Allocate data sets for CA Endevor SCM base and delta libraries for your Predict element types, if you are using CA Endevor SCM Integration for the Natural Environment to manage Predict types. Make sure that you use your site naming conventions. Define the PDS data sets with a RECFM value of VB and an LRECL value of at least 1796. All Predict element types can share a single pair of base and delta libraries.

3. Allocate list libraries. Make sure that you use your site naming conventions. Natural and Predict elements can use list libraries already defined for other element types, but you can define separate list libraries. We recommend that within any given CA Endevor SCM stage, all Natural elements and Predict elements use the same list library.

Edit, Reassemble, and Relink ENCOPTBL

To activate CA Endevor Integration for the Natural Environment, you must edit the Optional Features table (ENCOPTBL). As delivered, this option is turned off.

Follow these steps:

1. Find the ENCOPTBL in use at your site. This table is delivered in the *iprfx.iqual.*CSIQSRC library. To verify the table in use at your site, you can look in your site's C1DEFLTS table where the table should already be defined using the OPTTBL=parameter.
2. Find the following line in ENCOPTBL: ENHOPT EINE_ACTIVE=ON and uncomment the line. To find this option in the table, you can go to the command line and type F 'EINE_ACTIVE' and press Enter.
3. Use the JCL member BC1JTABL in the *iprfx.iqual*.CSIQJCL library delivered with the product to reassemble and relink ENCOPTBL.

   The ENHOPT_EINE ACTIVE parameter is now active in your site's ENCOPTBL This is a requirement to enable the CA Endevor Integration for the Natural Environment option.

Note: For more information on using the Optional Features Table, see the chapter "Using Optional Features" in the *Administration Guide.*

How to Add Site Symbolics to Associate Natural with CA Endevor SCM

The CA Endevor SCM site symbolics facility enables you define global symbols that can be used in type and processor definitions to reference data set name specifications for base, delta, source output, include libraries, and processors. This simplifies maintenance, because these commonly referenced data sets can be defined in a single location. Perform the following steps to add the site symbolics:

1. Define site symbolics.
   a. Define site symbolics to associate each CA Endevor SCM stage and Natural development system to a specific FUSER and FDIC combination. You can define a maximum of 99 sets of these site symbolics. For more information, see Associate Each CA Endevor SCM Stage to a Specific Natural FUSER/FDIC Combination (see page 223).
   b. Define site symbolics to associate your Natural libraries to CA Endevor SCM locations. For more information, see Methods of Associating Natural Libraries to CA Endevor SCM Inventory Locations (see page 226).

2. Use the JCL member BC1JSYMT to create the site symbolics table.
3. Define the site symbolics table in C1DEFLTS using the SYMBOLTBL= parameter to define the table name.

Note: For more information on adding site symbolics to CA Endevor SCM, see the chapter "Site Symbolics" in the *CA Endevor Software Change Manager Administration Guide*.

Associate Each CA Endevor SCM Stage to a Specific Natural FUSER/FDIC Combination

You must add a set of CA Endevor SCM Integration for the Natural Environment site symbolics to CA Endevor SCM for each distinct Natural FUSER/FNAT/FDIC set that is to be accessed by CA Endevor SCM. The CA Endevor SCM Integration for the Natural Environment site symbolics are used by the Natural and Predict processors and by the CA Endevor SCM Integration for the Natural Environment batch job submission facility. Because a stage is directly associated with a specific Natural FUSER/FNAT/FDIC set, each stage must have a corresponding set of CA Endevor SCM Integration for the Natural Environment site symbolics. Additionally, you must add a set of CA Endevor SCM Integration for the Natural Environment site symbolics for each specific Natural FUSER/FNAT/FDIC set associated with any development environment that needs to be accessed by CA Endevor SCM Integration for the Natural Environment for add or retrieve actions. These development Natural FUSER/FNAT/FDIC sets will not necessarily be associated with a stage.

To associate each CA Endevor SCM stage to a specific Natural FUSER/FDIC combination, define the following set of site symbolics for each distinct Natural FUSER/FNAT/FDIC set that is to be accessed by CA Endevor SCM. You can edit the template for $ESYMBOL definitions provided as JCL member SITESYMT and the sample set of $ESYMBOL definitions provided as JCL member SITESYME.

Each set of CA Endevor SCM Integration for the Natural Environment site symbolics must contain the following symbol names with all symbols in the set sharing a common symbol-set suffix:

SYMNAME=#FUSER#X,

SYMDATA=(*xxx,yyy)*

Where xxx is FUSER DBID and yyy is FUSER File number, for example, (002,029).

SYMNAME=#FDIC#X,

SYMDATA=(*xxx,yyy)*

Where xxx is FDIC DBID and yyy is FDIC File number, for example, (002,032).

SYMNAME=#ADALOAD#X,

SYMDATA=*Adabas_load_library*

Name of the Adabas load library used in batch access to the site symbolic set's FUSER and FDIC combination, for example, ADAvrs.LOAD.

SYMNAME=#NATLOAD#X,

SYMDATA=*Natural_load_library*

Name of the Natural load library used in batch access to the site symbolic set's FUSER and FDIC combination, for example, NATnnn.LOAD.

SYMNAME=#NATNUC#X,

SYMDATA=*Natural_batch_nucleus*

Name of the Natural batch nucleus used in batch access to the site symbolic set's FUSER and FDIC combination, for example, NATnnnBA.

SYMNAME=#NATPARM#X,

SYMDATA='*NATPARM values*'

Any site-specific NATPARM values used in batch access to the site symbolic set's FUSER and FDIC combination, for example:
PARM=PRM002BA,IM=D,INTENS=1,MAXCL=0,MADIO=0,MT=0

SYMNAME=#DDCARD#X,

SYMDATA='*DDCARD*'

A literal string of valid ADARUN parameters used in batch access to the site symbolic set's FUSER and FDIC combination, for example: ADARUN MODE=MULTI,SVC=221,DEVICE=8390,DBID=002.

This symbol name is optional if a value is defined for #DDCARDD#X.

For more information, see Adabas DDCARD Format (see page 225).

SYMNAME=#DDCARDD#X,

SYMDATA='*DDCARDD*'

The name of the data set used for either a PDS member or a PS file. The SYMDATA must be set to a null value if #DDCARD#X is specified.

SYMNAME=#DDCARDM#X,

SYMDATA='*DDCARDM*'

The name of the PDS member. The SYMDATA must be set to a null value if #DDCARD#X is specified or a PS file is specified in the #DDCARDD#X.

SYMNAME=#LISTLIB#X,

SYMDATA='*list_library*'

Name of the List library that is defined for elements in the stage that is associated with the above FDIC, FUSER combination, for example, CA.NDV.CASCM.TEST.NAT.LISTLIB. This symbol can be omitted if the CA Endevor SCM Integration for the Natural Environment Site Symbolic set is not related to a stage.

Adabas DDCARD Format

There are three different ways to specify the Adabas DDCARD information:

- An inline literal string such as ADARUN MODE=MULTI,…. If this format is used, then you must code #DDCARD#X and null values in SYMDATA for #DDCARDD#X and DDCARDM#X.
- A sequential dataset (PS). If this format is used, then you must code the data set name in #DDCARDD#X and null values in SYMDATA for #DDCARD#X and #DDCARDM#X.
- A member in a PDS. If this format is used, then you must specify the name of the PDS in #DDCARDD#X and the member name in #DDCARDM#X and set #DDCARD#X SYMDATA to null values.

Example: Site Symbolics Defined in BC1JSYMT

This example shows part of a BC1JSYMT JCL member that has been edited to include a typical CA Endevor SCM Integration for the Natural Environment site symbolics set.

```
$ESYMBOL SYMNAME=#FUSER#T,SYMDATA=(002,029)

$ESYMBOL SYMNAME=#FDIC#T,SYMDATA=(002,032)

$ESYMBOL SYMNAME=#ADALOAD#T,SYMDATA=ADAvrs.LOAD

$ESYMBOL SYMNAME=#NATLOAD#T,SYMDATA=NATnnn.LOAD

$ESYMBOL SYMNAME=#NATNUC#T,SYMDATA=NAT316BA

$ESYMBOL SYMNAME=#NATPARM#T,
 SYMDATA='PARM=PRM002BA,IM=D,INTENSX=1,MAXCL=0,MADIO=0,MT=0'

$ESYMBOL SYMNAME=#DDCARD#T,SYMDATA=''

$ESYMBOL SYMNAME=#DDCARDD#T,SYMDATA=Adabas.SOURCE

$ESYMBOL SYMNAME=#DDCARDM#T,SYMDATA=DDCARD

$ESYMBOL SYMNAME=#LISTLIB#T,
 SYMDATA='CA.NDV.CASCM.TEST.NAT.LISTXLIB'
```

Methods of Associating Natural Libraries to CA Endevor SCM Inventory Locations

Your Natural libraries must be mapped to CA Endevor SCM inventory locations using site symbols. Select one of the following methods depending on whether your Natural library names follow a naming convention:

- **Direct method:** If your Natural library names follow a naming convention, you can use a direct method of symbolic substitution. For more information, see Associate Natural Library Names that Follow a Naming Convention to CA Endevor SCM Inventory Locations (see page 226).
- **Indirect method:** If your Natural Libraries, do not follow a naming convention, you can use an indirect method that uses recursive symbolic substitution. We recommend this method for sites that already have CA Endevor SCM configured. For more information, see Associate Any Natural Library Names to CA Endevor SCM Inventory Locations (see page 228).

Both methods require the following symbols:

SYMNAME=#NSRCLIB,

SYMDATA='source_Natural_library'

Name of the source Natural library from which Natural objects will be moved. The symbolic uses the following CA Endevor SCM symbolics to identify the source Natural library.

- &C1SENVMNT(1,1) – first character of the environment name
- &C1SSTGNUM – stage number (1 or 2)
- &C1SSUBSYS – subsystem name

SYMNAME=#NTGTLIB,

SYMDATA='target_Natural_library'

Name of the target Natural library when the following actions are performed: delete, generate, or move. The symbolic uses the following CA Endevor SCM symbolics to identify the target Natural library.

- &C1EN(1,1) – first character of the environment name
- &C1S# – stage number (1 or 2)
- &C1SU – subsystem name

Associate Natural Libraries that Conform to a Naming Standard to CA Endevor SCM Inventory Locations

You can map your Natural libraries to CA Endevor SCM subsystems using a direct method of symbolic substitution, provided you use a standard naming convention for your Natural libraries. The following method lets you define your Natural libraries to CA Endevor SCM locations, using standard CA Endevor SCM location symbolics.

To associate Natural libraries that conform to a naming standard to CA Endevor SCM Inventory Locations

Define the symbolics #NTGTLIB and #NSRCLIB, to assign Natural libraries that conform to a standard naming convention, using one of the formats shown next--either the environment, stage number and subsystem, or the subsystem only.

- Define the symbolics using the environment, stage number and subsystem:

```
$ESYMBOL SYMNAME=#NTGTLIB,
     SYMDATA='&&C1EN(1,1)&&C1S#&&C1SU'
$ESYMBOL SYMNAME=#NSRCLIB,
     SYMDATA='&&C1SENVMNT(1,1)&&C1SSTGNUM&&C1SSUBSYS'
```

- Define the symbolics using the subsystem only:

```
$ESYMBOL SYMNAME=#NTGTLIB,
     SYMDATA='&&C1SU'
$ESYMBOL SYMNAME=#NSRCLIB,
     SYMDATA='&&C1SSUBSYS'
```

Depending upon which format you code, the names of the Natural libraries are assigned directly by the symbol values using either the first character of environment name with the stage number and subsystem name, or using the subsystem name alone. No additional site symbols are required.

Example: Associate Natural Libraries to CA Endevor SCM Inventory Locations Using Direct Symbolic Substitution

This example uses a direct method of symbolic substitution to associate Natural library names to CA Endevor SCM locations and assumes that the site's lifecycle is configured as follows:

- It has two environments (QA and PRD) and four stages (two stages in each environment) in the following order:

  QA/1 -> QA/2 -> PRD/1 -> PRD/2

- The subsystem name is NSUB for all stages.

Define the following site symbolics in your site symbol table (double ampersands are required in the site symbol table):

```
$ESYMBOL SYMNAME=#NTGTLIB,
     SYMDATA='&&C1EN(1,1)&&C1S#&&C1SU'
$ESYMBOL SYMNAME=#NSRCLIB,
     SYMDATA='&&C1SENVMNT(1,1)&&C1SSTGNUM&&C1SSUBSYS'
```

Therefore, the symbolics resolve to the following libraries:

- At QA stage 1, &#NTGTLIB and &#NSRCLIB resolve to Q1NSUB as your Natural library name.
- At QA stage 2, &#NTGTLIB and &#NSRCLIB resolve to Q2NSUB as your Natural library name.
- At PRD stage 1, &#NTGTLIB and &#NSRCLIB resolve to P1NSUB as your Natural library name.
- At PRD stage 2, &#NTGTLIB and &#NSRCLIB resolve to P2NSUB as your Natural library name.

Associate Any Natural Libraries to CA Endevor SCM Inventory Locations

You can map your Natural libraries to CA Endevor SCM inventory locations using a indirect method of symbolic substitution, if your Natural library names do not follow a standard naming convention. The following method lets you define your Natural libraries to CA Endevor SCM locations, using standard CA Endevor SCM location symbolics and recursive symbol substitution.

To associate any Natural library names to CA Endevor SCM inventory locations

1. Define the symbolics #NTGTLIB and #NSRCLIB using recursive symbol substitution. You can use the following $ESYMBOL definitions provided as JCL member SITESYME to associate any Natural library name to a CA Endevor SCM location.

   ```
   $ESYMBOL SYMNAME=#NTGTLIB,
        SYMDATA='&&#N&&C1EN(1,1)&&C1S#&&C1SU'
   $ESYMBOL SYMNAME=#NSRCLIB,
        SYMDATA='&&#N&&C1SENVMNT(1,1)&&C1SSTGNUM&&C1SSUBSYS'
   ```

   The #NSRCLIB symbolic indirectly defines the source Natural library from which Natural objects will be moved. The #NTGTLIB symbolic indirectly defines the target Natural library when the following actions are performed: delete, generate, or move. The symbols resolve to a series of other symbols, defined using the #N*xysubsys* symbolic, which you must also code in your site symbol table.

2. Define one #N*xysubsys* symbolic for each stage and subsystem combination that you want to control using CA Endevor SCM.

   ```
   $ESYMBOL SYMNAME=#Nxysubsys,
        SYMDATA='Natural_library_name'
   ```

   #N*xysubsys*

   - x – first character of the environment
   - y – stage number (1 or 2)
   - subsys – subsystem name

The symbolics #NSRCLIB and #NTGTLIB, when set using recursive symbol substitution, resolve to the series of symbols you define using the #N*xysubsys* symbolic. The *#Nxysubsys* symbolic defines your Natural libraries, which can reside in any combination of one to *n* FUSER files, where *n* is the number of stages in your lifecycle. For more information, see Associate Each CA Endevor SCM Stage to a Specific Natural FUSER/FDIC Combination (see page 223).

Example: Associate Natural Libraries to CA Endevor SCM Inventory Locations Using Indirect Symbolic Substitution

This example uses an indirect method of symbolic substitution to associate Natural library names to CA Endevor SCM locations, assuming that your site's lifecycle is configured as follows:

- It has two environments (QA and PRD) and four stages (two stages in each environment) in the following order:

  QA/1 -> QA/2 -> PRD/1 -> PRD/2

- The subsystem name is NSUB for all stages.
- The library names are **not** the same.

Define the following site symbolics in your site symbol table (double ampersands are required in the site symbol table):

```
$ESYMBOL SYMNAME=#NTGTLIB,
     SYMDATA='&&#N&&C1EN(1,1)&&C1S#&&C1SU'
$ESYMBOL SYMNAME=#NSRCLIB,
     SYMDATA='&&#N&&C1SENVMNT(1,1)&&C1SSTGNUM&&C1SSUBSYS'
$ESYMBOL SYMNAME=#NQ1NSUB,
     SYMDATA='NATLIB1'
$ESYMBOL SYMNAME=#NQ2NSUB,
     SYMDATA='NATLIB2'
$ESYMBOL SYMNAME=#NP1NSUB,
     SYMDATA='NATLIB3'
$ESYMBOL SYMNAME=#NP2NSUB,
     SYMDATA='NATLIB4'
```

Coding these symbolics, has the following effects:

- The target and source library symbolics #NTGTLIB and #NSRCLIB resolve to the following stage symbols:

  Note: #N is not defined as a site symbol so no substitution takes place.

  - &#NQ1NSUB
  - &#NQ2NSUB
  - &#NP1NSUB
  - &#NP2NSUB

- The value of the secondary symbols resolve to the assigned Natural library name:
  - #NQ1NSUB=NATLIB1
  - #NQ2NSUB=NATLIB2
  - #NP1NSUB=NATLIB3
  - #NP2NSUB=NATLIB4
- Therefore, the symbolics resolve to the following libraries:
  - At QA/1, &#NTGTLIB and &#NSRCLIB resolve to NATLIB1 as your Natural library name.
  - At QA/2, &#NTGTLIB and &#NSRCLIB resolve to NATLIB2 as your Natural library name.
  - At PRD/1, &#NTGTLIB and &#NSRCLIB resolve to NATLIB3 as your Natural library name.
  - At PRD/2, &#NTGTLIB and &#NSRCLIB resolve to NATLIB4 as your Natural library name.

Install the Processors

CA Endevor SCM uses JCL streams called processors to automate the creation of executables such as program load modules. You must associate one or more processor groups with each type of source code. Each processor group identifies the processors needed for a particular type of source. User and site symbolics make it possible to share processors across groups. You must install the Natural and Predict Generate, Delete, and Move processors.

As delivered, the software uses the Natural SYSOBJH utility for storing objects in CA Endevor SCM. However, you can configure the software to use the utility SYSTRANS instead of SYSOBJH.

Use one of the following methods to install the processors and configure the software:

- Install the Processors to Use the Natural SYSOBJH Utility (see page 231)
- Install the Processors to Use the Natural SYSTRANS Utility (see page 231)

Install the Processors to Use the Natural SYSOBJH Utility

This method installs and configures your processors to use the Natural SYSOBJH utility for storing objects in CA Endevor SCM.

To install and configure the Natural and Predict Generate, Delete, and Move processors:

1. Edit the sample JCL member DEFPROC to conform to your site's standards. Edit the job card and data set names. You may also need to edit the ENVIRONMENT and SYSTEM parameters within each ADD ELEMENT SCL statement.
2. (Required) Modify the Generate processor, PNATGEN, to set the UTILOPT parameter to match what the +EINE-SYSTRAN-SYSOBJH parameter in the initialization exit, UEXIT00, is set to. For more information, see the step Modify the Initialization Exit in Install User Exits (see page 209).
3. (Optional) Modify the processors replacing the AUTO=ON parameter with the appropriate Processor Natural Security logon and password, if you use Natural Security.

   Note: If you use Natural Security, two different Natural Security logon IDs are required: one for the processors that uses a default ETID of *INIT-US, so the job name is used as the ETID, and a second for the CASCMMNI server that uses a default ETID of *INIT-ID, so the server uses the step name as the ETID. It is critical that the processors use a different ETID from the server, because they are accessing Natural libraries from the server job at the same time.
4. Submit the DEFPROC JCL.

   All the integration software's Natural objects are loaded to Natural.

Install the Processors to Use the Natural SYSTRANS Utility

This method installs and configures your processors to use the Natural SYSTRANS utility for storing objects in CA Endevor SCM.

To install and configure the Natural and Predict Generate, Delete, and Move processors:

1. Modify the following processors in the EINE JCL library:
   - Delete PNATGEN and PNATMOV.
   - Rename PNATGENT and PNATMOVT to PNATGEN and PNATMOV, that is, remove the T suffix.

   This eliminates the need to edit the CA Endevor SCM setup member DEFPROC for these files.
2. Edit the sample JCL member DEFPROC to conform to your site's standards. Edit the job card and data set names. You may also need to edit the ENVIRONMENT and SYSTEM parameters within each ADD ELEMENT SCL statement.

3. (Optional) Modify the processors replacing the AUTO=ON parameter with the appropriate Processor Natural Security logon and password, if you use Natural Security.

   Note: If you use Natural Security, two different Natural Security logon IDs are required: one for the processors that uses a default ETID of *INIT-US, so the job name is used as the ETID, and a second for the CASCMMNI server that uses a default ETID of *INIT-ID, so the server uses the step name as the ETID. It is critical that the processors use a different ETID from the server, because they are accessing Natural libraries from the server job at the same time.

4. Submit the JCL, after modifying DEFPROC.

   All the integration software's Natural processors are loaded to Natural and the integration software is set up to use the Natural SYSTRANS utility for storing objects in CA Endevor SCM.

5. Log on to the EINELIB Natural library. Edit the UEXIT00 startup user exit module near the bottom to uncomment the following MOVE statement and change 'O' to 'T' as follows:

   ```
   MOVE 'T' TO +EINE-SYSTRANS-SYSOBJH
   ```

6. STOW the exit.

7. Rename the SYSTRANS TEXT and SYSOBJH TEXT members in the EINELIB Natural library, as follows:

   a. Delete the TEXT members:

      NATADD

      NATDADD

      NATDRETR

      NATEADD

      NATRETR

   b. Rename the SYSTRANS TEXT members to use the original names:

NATADDT	to	NATADD
NATDADDT	to	NATDADD
NATDRETT	to	NATDRETR
NATEADDT	to	NATEADD
NATRETRT	to	NATRETR

   The integration software is set up to use the Natural SYSTRANS utility for storing objects in CA Endevor SCM.

Define the Natural and Predict Types

Define your Natural and Predict Element types, processor groups, and processor group symbolics for each stage and system that is being configured for management of Natural and Predict elements. All subsystems defined to a system can use the types defined to that system.

To define the Natural and Predict types

1. Modify sample JCL member DEFTYPE, making sure that the job card, STEPLIB, and CONLIB definitions adhere to site standards.
2. Make a copy of DEFTYPE for each stage and system combination that is being configured for management of Natural elements. For example, if you have six stages in your Natural lifecycle, you need six copies of DEFTYPE.

   Copies of DEFTYPE are now available so you can edit them separately with stage-specific information.
3. Modify each copy of member DEFTYPE with the appropriate environment, stage, and system parameters; the base and delta libraries; and the symbol definitions. There are many definition sets within DEFTYPE (one for each Natural element type and Predict element type), and each needs to be assigned identical values for all of the following parameters and symbol definitions; therefore, we recommend that you use the CHANGE ALL edit to speed and simplify this edit procedure. Modify the italicized values in the following lines of DEFTYPE.

   - Environment, stage, and system parameters

     TO ENV *'ENVNAME'*

     STAGE ID *'Z'*

     CA Endevor SCM environment and stage

     SYSTEM *'SYSNAME'*

     CA Endevor SCM system for which the types, processor groups, and Symbolics are being defined.

   - Base and delta libraries

     All Natural element types can share the same base and delta libraries. All Predict element types can also share the same base and delta libraries.

BASE LIBRARY *'NAT.SRCLIB'*

CA Endevor SCM base Library for Natural element types

DELTA LIBRARY *'NAT.DELTA'*

CA Endevor SCM delta Library for Natural element types

BASE LIBRARY *'PRED.SRCLIB'*

CA Endevor SCM base Library for Predict element types

DELTA LIBRARY *'PRED.DELTA'*

CA Endevor SCM delta Library for Predict element types

- Symbol definitions

  All symbol definitions reference the CA Endevor SCM Integration for the Natural Environment site symbolic sets.

  - All processor groups (generate, delete, and move) share the same set of symbols, as follows.

    ```
    SYMBOL ADALOAD=&#ADALOAD#X
    SYMBOL NATLOAD=&#NATLOAD#X
    SYMBOL NATNUC=&#NATNUC#X
    SYMBOL NATPARM=&#NATPARM#X
    SYMBOL DDCARD=&#DDCARD#X
    SYMBOL DDCARDD=&#DDCARDD#X
    SYMBOL DDCARDM=&#DDCARDM#X
    SYMBOL LISTLIB=&#LISTLIB#X
    ```

    Use the CHANGE ALL command to replace the #X in the symbol definitions to the value used in the CA Endevor SCM Integration for the Natural Environment site symbolic set that corresponds to the stage for which the current copy of DEFTYPE is being configured. For example, if you set the stage ID parameter to STAGE ID = T, then you would change the #X in the symbol definitions to #T, assuming that this is the suffix used in the CA Endevor SCM Integration for the Natural Environment site symbolic set that corresponds to stage T.

- In addition, all move processor groups also share the following additional set of symbols.

```
SYMBOL ADALOAD2=&#ADALOAD#Y
SYMBOL NATLOAD2=&#NATLOAD#Y
SYMBOL NATNUC2=&#NATNUC#Y
SYMBOL NATPARM2=&#NATPARM#Y
SYMBOL DDCARD2=&#DDCARD#Y
SYMBOL DDCARDD2=&#DDCARDD#Y
SYMBOL DDCARDM2=&#DDCARDM#Y
SYMBOL LISTLIB2=&#LISTLIB#Y
```

  Use the CHANGE ALL command to change the #Y in the symbol definitions to the value used in the CA Endevor SCM Integration for the Natural Environment site symbolic set that corresponds to the stage to which elements will be moved. For example, if elements will be moved from stage T (test) to stage Q (QA), then you would change the #Y suffix to #Q (assuming that this is the suffix used in the CA Endevor SCM Integration for the Natural Environment Site Symbolic set that corresponds to stage Q).

4. Submit the JCL for each copy of DEFTYPE (once for each stage and system combination) created and edited in the previous step.

Enable Global Type Sequencing

To use the CA Endevor SCM Global Type Sequencing option to process Natural types, you must enable this option for the Natural types.*Global Type Sequencing* is an optional method of defining the order in which element actions are processed. When global type sequencing is enabled, element actions are processed by type sequence regardless of the inventory location of each action. API element actions (if wildcarded) and SCL element actions are executed in the type sequence order defined at the site level in the Type Sequence member created by the CA Endevor SCM administrator. The Global Type Sequencing option is required to specify the order in which your Natural types are processed in batch requests.

To enable Global Type Sequencing for Natural types

1. Enable Global Type Sequencing, if it is not already in use at your site.

   Note: For more information on enabling Global Type Sequencing, see the scenario "How to Enable Global Type Sequencing" in the *Scenario Guide*.

2. Edit your Global Type Sequencing Type Processing Member to add the Natural types. You must use the following list of Natural types.

Important! Do not change the order in which the types are listed in the supplied list of Natural types.

The following list of Natural types is supplied as JCL member ETYPESEQ.

```
TYPE 'PREDFI'  DESC 'Predict TYPE DEF FOR FILE'.
TYPE 'NTDDM'   DESC 'Natural TYPE DEF FOR DDM OBJECTS'.
TYPE 'NATGDA'  DESC 'Natural TYPE DEF FOR GLOBALS DATA AREAS'.
TYPE 'NATPDA'  DESC 'Natural TYPE DEF FOR PARAMETER DATA AREAS'.
TYPE 'NATLDA'  DESC 'Natural TYPE DEF FOR LOCAL DATA AREAS'.
TYPE 'NATCOPY'  DESC 'Natural TYPE DEF FOR COPYCODE'.
TYPE 'NATTEXT'  DESC 'Natural TYPE DEF FOR TEXT'.
TYPE 'NATSUBR'  DESC 'Natural TYPE DEF FOR SUBROUTINES'.
TYPE 'NATSUBP'  DESC 'Natural TYPE DEF FOR SUBPROGRAMS'.
TYPE 'NATHELP'  DESC 'Natural TYPE DEF FOR HELPROUTINES'.
TYPE 'NATMAP'   DESC 'Natural TYPE DEF FOR MAP'.
TYPE 'NATPROG'  DESC 'Natural TYPE DEF FOR PROGRAMS'.
TYPE 'NATCLASS' DESC 'Natural TYPE DEF FOR CLASSES'.
```

Your Natural types are processed in the order in which the types are listed in the Type Processing Member.

Customize the Server Start JCL

To enable the server to be turned on, you must customize the CA Endevor SCM Integration for the Natural Environment server JCL to your site specifications.

Note: The servers can be executed as a batch job using the provided JCL member or you can modify it to run as a started task and copy it to a z/OS procedure library such as SYS1.PROCLIB.

To customize the server start JCL

1. Modify the sample JCL member CASCMMN1 as follows:
   a. Make sure that the job card adheres to site standards. The job card in the MSGLEVEL is set to (0,0) to suppress extensive informational output from the CA Endevor SCM API.
   b. Set appropriate values for the PREFIX and SERVER symbolic variables, which serve to establish the names of the data sets allocated and used by the started task.
   c. In the CASCMMN1 step, make the following modifications:

      Important! If this is a Natural batch nucleus which is also defined in any of your CA Endevor SCM Integration for the Natural Environment site symbolics as a #NATNUC value, then this Natural batch nucleus must be linked as reentrant.

      - Modify the single PARM value to a Natural batch nucleus which provides access to the Natural environment (FUSER/FNAT/FDIC) in which the CA Endevor SCM Integration for the Natural Environment online sessions will be running.
      - Make sure that the STEPLIB data sets adhere to site standards. Make sure that the Natural load library you specify contains the Natural batch nucleus.
      - In the Natural/Adabas DATA SET section, make sure that the DDCARD data set is properly defined, either inline or with reference to an external DDCARD data set.
2. (Optional) Make copies of the JCL member and alter the job name and the SERVER symbolic variable appropriately, if you want to configure and run multiple servers.

3. (Optional) Modify the server JCL to provide the appropriate server Natural Security login and password, if you use Natural Security.

   Note: If you use Natural Security, two different Natural Security logon IDs are required: one for the processors that uses a default ETID of *INIT-US, so the job name is used as the ETID, and a second for the CASCMMNI server that uses a default ETID of *INIT-ID, so the server uses the step name as the ETID. It is critical that the processors use a different ETID from the server, because they are accessing Natural libraries from the server job at the same time.

4. Set up each configured JCL member as a started task, by copying it to a z/OS procedure library such as SYS1.PROCLIB.

Customize the Server Shutdown JCL

A CA Endevor SCM Integration for the Natural Environment server is shutdown through submission of a separate started task or batch job that issues a shutdown command to the server task using the message queue. Each server task requires a corresponding shutdown task.

To customize the server shutdown JCL

1. Modify sample JCL member CASCMSTP as follows:

   a. Make sure that the JOBLIB data sets adhere to site standards. The Natural load library should be that in which the Natural batch nucleus is found.

   b. In the SHUTDOWN step, make the following modifications:

      - Modify the single PGM value to a Natural batch nucleus which provides access to the Natural environment (FUSER/FNAT/FDIC) in which CA Endevor SCM Integration for the Natural Environment online sessions will be running. This is usually the same Natural batch nucleus as that used in the corresponding CASCMMN1 server started task.
      - Make sure that the DDCARD data set is properly defined, either inline or with reference to an external DDCARD data set.
      - In the CMSYNIN section, make sure that the single parameter submitted to the SHUTDOWN program is the name of the server started task that is to be shutdown. For example, to shut down a server started task called CASCMMN1, the command should be:

        ```
        SHUTDOWN CASCMMN1
        ```

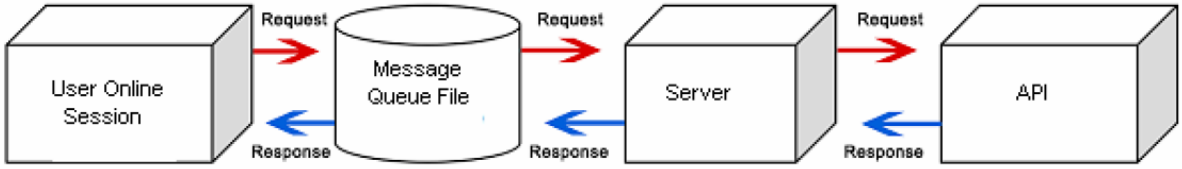
   If you are configuring and running multiple server instances, a separate copy of this JCL member should be made, with job name and SHUTDOWN command modified appropriately.

2. Set up each configured member as a started task, by copying it to a z/OS procedure library such as SYS1.PROCLIB.

How Processing Works

CA Endevor SCM Integration for the Natural Environment operates in a client/server fashion to provide an interface between user online sessions and CA Endevor SCM. Requests are processed in the following manner:

- Each user online session does not directly invoke the CA Endevor SCM API. Instead, it writes an API request to the message queue file, from which the API request is read by a started task referred to as the server.
- The server submits the API request to CA Endevor SCM, receives the API response, and writes it to the message queue.
- The user online session reads the API response from the message queue and provides appropriate output to the user. The following flowchart shows how user online session requests are processed in CA Endevor SCM Integration for the Natural Environment.



Servers and Session Response Times

Each server started task is a single-threaded processor. For multi-threaded processing, multiple servers can be configured and run simultaneously. The optimal number of servers required at a site depends on the number of users and the frequency with which they simultaneously run CA Endevor SCM Integration for the Natural Environment online sessions.

The server is a stateless server, meaning that it treats each request as an independent transaction that is unrelated to any previous request. Each request is processed on a first-in-first-out basis without regard to which user submitted it. Due to the nature of the server processing, a dedicated server task is not required for each user.

There is no limit to the number of users a single server can support. If 100 users are running concurrent sessions, all of their list requests and action requests are processed. However, users do experience a relatively slower response to their requests than if the concurrent sessions were significantly fewer in number. The response time can vary widely from minute to minute, depending upon whether requests are submitted simultaneously. For example, if all 100 users never submit a request at the same time that another user's request is being processed, then there is no delay in processing. In this case, it is as if each user has the server all to themselves.

If your site has multiple concurrent users who believe that their response time is unacceptably slow and a review of the server logs shows that a large number of requests are regularly processed in close succession during certain times (suggesting that multiple requests are often simultaneously queued up), then you may want to improve session response times. To do this you can alter the class or priority of the server task, add additional server tasks, or both.

Administrators can change the server and API timeout values in the startup user exit, UEXIT00. The timeout values limit the time for the server to begin processing a request and for the API to complete processing of the request. If the timeout intervals are exceeded, control is returned to the user with the option of continuing to wait for the request to complete or return to CA Endevor SCM Integration for the Natural Environment. If the user chooses not to wait for the request to complete, the request is not terminated. The server continues to wait for the request to complete, because there is no way for it to cancel a request sent to the API after it has begun. All records associated with the request are automatically cleared from the message queue file at a later time.

Server Operational Guidelines

The following guidelines apply to operating the CA Endevor SCM Integration for the Natural Environment Server:

- Although the server can be run as a long-running batch job, the server typically runs as a z/OS started task.
- There is no need to periodically shut down the server unless Adabas requires shut down.
- Before starting a server, ensure Adabas is up and running.
- If Adabas is shut down while the server is up, the server should be shut down and restarted after Adabas comes back up.

How to Startup the Server

If configured as a started task, start the server using the S command and the server name.

Example: Server Startup Command

A server named CASCMMN1 is started with the following operator command:

```
S CASCMMN1
```

How to Shutdown the Server

To shut down a server, submit its corresponding shutdown task (or batch job).

Example: Server Shutdown Command

If a server task has a corresponding shutdown task named CASCMSTP, then the shutdown task is started with the following operator command:

```
S CASCMSTP
```

Server Logs

You can review startup and API transaction information in the following server logs.

Server Startup Log

At startup time, the server writes startup information to data set CMPRT25, which may be useful in diagnosing configuration problems. This output includes a report on all site symbolics as extracted from the EN$TROPT report.

API Transaction Log

The server writes a summary of all API transactions processed to data set CMPRT01.

Example: API Transaction Log – CMPRT01

```
*** 2006-11-17 10:57:31.5 *** API function: EADQ ; User: MXJ1 ; Return code: 0000 ; Reason: 0000 ;
Hi Msg ID: AP00000I
*** Request ID: D4E7D1F1404040400633308936805F
*
*** 2006-11-17 11:07:34.6 *** API function: LELQ ; User: MXJ1 ; Return code: 0000 ; Reason: 0000 ;
Hi Msg ID: AP00000I
*** Request ID: D4E7D1F1404040400633308944507F
```

How to Convert from E/Nat

Sites that are converting from E/Nat to CA Endevor SCM Integration for the Natural Environment should perform these steps for a successful conversion.

To convert your E/Nat installation to CA Endevor SCM Integration for the Natural Environment, perform the following steps:

1. Install the CA Endevor SCM Integration for the Natural Environment software according to the installation instructions provided in the CA Endevor SCM Integration for the Natural Environment Implementation Guide.
2. To maintain Element Signout Status, execute the CA Endevor SCM Element Signed Out Profile - by System (Report number 8) for all inventory locations from which E/Nat elements will be added as CA Endevor SCM Integration for the Natural Environment Elements. Save the output.

   You will use this report later to set the signout to status on the elements you add to CA Endevor SCM Integration for the Natural Environment.
3. Add Elements for each Environment, Stage and E/Nat Type from your E/Nat Natural database directly to CA Endevor SCM Integration for the Natural Environment, using the CA Endevor SCM Integration for the Natural Environment Batch (or Foreground) Add facility.

   The target location of the Add Action should be the Environment and Stage that corresponds to the location associated with the Natural database. If the target location is Stage 2 of the Environment and that stage is not defined as an Entry Stage, then, after the Add Action, move the Elements to Stage 2.

   If you wish to ADD the Elements directly back to Stage 2 of the Environment, you can temporarily update the ENTRYSTG parameter specification in the C1DEFLTS Environment Definition for the Target Environment in Stage 2.

   If you do not want to overlay the Natural objects, set the “Generate (Catalog) Element” option to “N” on the Add/Update Natural Programming Objects screen before submitting the ADD request.
4. Using the Element Signed Out Profile - by System report, create and execute CA Endevor SCM SIGNIN actions (using the SIGNOUT TO Option clause) to properly set the Signout Status of the CA Endevor SCM Integration for the Natural Environment inventory.

   This sets the Signout to Status for each Element to match the status associated with each Element prior to adding it to CA Endevor SCM Integration for the Natural Environment.

5. Define an ESI (External Security Interface) rule to prohibit actions that can modify E/Nat Elements.

   This prevents changes to the E/Nat Inventory and retains the change history associated with the Elements prior to moving them to CA Endevor SCM Integration for the Natural Environment.

Note: If you prefer to transfer the E/Nat to CA Endevor SCM Integration for the Natural Environment in Logical Units, you can use the E/Nat Retrieve Action to extract the Element Source, on a CA Endevor SCM system/subsystem basis, from the E/Nat Inventory to a temporary repository. Use the CA Endevor SCM Integration for the Natural Environment ADD action to populate the new CA Endevor SCM Integration for the Natural Environment inventory area from the temporary repository. Delete the Elements from the temporary repository after they have been added to the CA Endevor SCM Integration for the Natural Environment inventory area.

Appendix G: Japanese Panels and Messages

For the Japanese installation, additional steps are needed to install the Japanese menus and panels. You must perform one of the following procedures depending on which method you are using to identify the CA Endevor SCM libraries to the ISPF environment.

- If you are using the ISPF LIBDEF Services method (see page 76), you must edit the ENDEVOR member (see page 246) to include the Japanese panel and menu library names in the appropriate data set concatenations.
- If you are using the TSO FREE and ALLOCATE method (see page 78), you must edit the BC1PCLS1 member (see page 247) to include the Japanese panel and menu library names in the appropriate data set concatenations.

This section contains the following topics:

Edit ENDEVOR Member for Japanese Installation

For the Japanese installation, the names of the libraries containing the Japanese ISPF messages and panels must be added to the ISPMLIB and ISPPLIB data set concatenations in the CLIST that you use to invoke the CA Endevor SCM product.

To Edit ENDEVOR Member

1. Make a backup copy of the CSIQCLS0 member ENDEVOR.

   This copy will be available if you need to view or use the original version of the member.

2. Edit the ENDEVOR member, using your site's ISPF editor, to change the following lines to look like the *After* example.

   Before (as delivered):

   ```
   ISPEXEC LIBDEF ISPPLIB DATASET ID('IPRFX.IQUAL.CSIQPENU')
   ISPEXEC LIBDEF ISPMLIB DATASET ID('IPRFX.IQUAL.CSIQMENU')
   ```

   After:

   ```
   ISPEXEC LIBDEF ISPPLIB DATASET ID +
   ('IPRFX.IQUAL.CSIQPJPN' +
    'IPRFX.IQUAL.CSIQPENU')
   ISPEXEC LIBDEF ISPMLIB DATASET ID +
   ('IPRFX.IQUAL.CSIQMJPN' +
    'IPRFX.IQUAL.CSIQMENU')
   ```

   The ENDVOR member has been edited to add the Japanese messages and panel libraries to the ENDEVOR CLIST library concatenations.

Edit Member BC1PCLS1 for Japanese Installation

For the Japanese installation, the names of the libraries containing the Japanese ISPF messages and panels must be added to the ISPMLIB and ISPPLIB data set concatenations in the CLIST that you use to invoke the CA Endevor SCM product.

To Edit BC1PCLS1 Member

1. Make a backup copy of the CSIQCLS0 member BC1PCLS1.

   This copy will be available if you need to view or use the original version of the member.

2. Edit the BC1PCLS1 member, using your site's ISPF editor, to change the following lines to look like the *After* example.

   Before (as delivered):

```
FREE  FI(ISPPLIB)
ALLOC FI(ISPPLIB) +
      DA('ISP.SISPPENU' +
         'IPRFX.IQUAL.CSIQPENU') +
      SHR

FREE  FI(ISPMLIB)
ALLOC FI(ISPSMLIB) +
      DA('ISP.SISPMENU' +
         'IPRFX.IQUAL.CSIQMENU') +
      SHR
```

   After:

```
FREE  FI(ISPPLIB)
ALLOC FI(ISPPLIB) +
      DA('ISP.SISPPENU' +
         'IPRFX.IQUAL.CSIQPJPN' +
         'IPRFX.IQUAL.CSIQPENU') +
      SHR

FREE  FI(ISPMLIB)
ALLOC FI(ISPMLIB) +
      DA('ISP.SISPMENU' +
         'IPRFX.IQUAL.CSIQMJPN' +
         'IPRFX.IQUAL.CSIQMENU') +
      SHR
```

   The BC1PCLS1 member has been edited to add the Japanese messages and panel libraries to the ENDEVOR CLIST library concatenations.

Appendix H: Installation Data Set Names

The data set names delivered with Version 15.0 and above are different than those delivered with releases earlier than CA Endevor SCM r14. Beginning with r14, the data set names were changed to make CA Endevor SCM compatible with CA CSM.

The following tables identify the old and new files names. These tables are provided in case you are familiar with the old names from prior installations.

CA Endevor SCM Data Set Names

Old Target DSN (Prior to r14)	New Target DSN (Beginning with r14)
Iprfx.iqual.AUTHLIB	Iprfx.iqual.CSIQAUTH
uprfx.uqual.AUTHLIB	iprfx.iqual.CSIQAUTU
iprfx.iqual.CONLIB	iprfx.iqual.CSIQLOAD
iprfx.iqual.OBJLIB	iprfx.iqual.CSIQMOD
iprfx.iqual.ISPPLIB	iprfx.iqual.CSIQPENU
iprfx.iqual.ISPMLIB	iprfx.iqual.CSIQMENU
iprfx.iqual.ISPTLIB	iprfx.iqual.CSIQTENU
iprfx.iqual.SOURCE	iprfx.iqual.CSIQOPTN
iprfx.iqual.ISRCLIB	iprfx.iqual.CSIQCLS0
iprfx.iqual.ISPSLIB	iprfx.iqual.CSIQSENU
iprfx.iqual.JCLLIB	iprfx.iqual.CSIQJCL
iprfx.iqual.RIJCL	iprfx.iqual.CSIQJCLR
iprfx.iqual.RIPROC	iprfx.iqual.CSIQPRCR
iprfx.iqual.SAMPLE	iprfx.iqual.CSIQSAMP
iprfx.iqual.TABLES	iprfx.iqual.CSIQSRC

CA Endevor CA Roscoe Interface Data Set Names

Product Component	Old Target DSN (Prior to r14)	New Target DSN (Beginning with r14)
ROSCOE Modules	iprfx.iqual.ROSCOE.OBJLIB	iprfx.iqual.CSIQMOD
ROSCOE Panels	iprfx.iqual.ROSCOE.PNLLIB	iprfx.iqual.CSIQPENU
ROSCOE Source	iprfx.iqual.ROSCOE.ROSLIB	iprfx.iqual.CSIQOPTN
ROSCOE Macros	iprfx.iqual.ROSCOE.MACLIB	iprfx.iqual.CSIQOPTN

CA Endevor SCM Japanese Panels and Messages Data Set Names

Product Component	Old Target DSN (Prior to r14)	New Target DSN (Beginning with r14)
Japanese Messages	iprfx.iqual.ISPMJPN	iprfx.iqual.CSIQMJPN
Japanese Panels	iprfx.iqual.ISPPJPN	iprfx.iqual.CSIQPJPN

Index